UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITECNICA SUPERIOR

DEPARTAMENTO DE TEORÍA DE LA SEÑAL Y COMUNICACIONES

PROYECTO FIN DE CARRERA

PREDICTOR DE CANCIONES DE ÉXITO

INGENIERÍA TÉCNICA DE TELECOMUNICACIONES: SONIDO E IMAGEN

Autor: Joyce Crossley Tavera

Tutor: Emilio Parrado Hernández

Leganés

Resumen

En este documento se presenta un método de predicción de canciones de éxito basado en el post-procesado de señal de audio y algoritmos de regresión.

Con la ayuda de la muestra de más de 180 canciones de diferentes géneros, aunque en su mayoría pop, se ha diseñado un predictor con un porcentaje de error relativamente bajo.

Gracias a este estudio se definen las características más relevantes a la hora de decidir si una canción será un éxito y en qué grado. También se comprobará si la propuesta sirve para cualquier otro género musical. En este caso se emplean canciones de género rock y se aplica el modelo elegido, obteniendo el error de predicción y comparándolo con el de pop.

Agradecimientos

Una mención especial a Emilio.

A la universidad.

¡Gracias!

Índice

1

Introducción

Desde la introducción de las tecnologías del audio digital hace más de 35 años, los avances computacionales y el tratamiento de señal han mejorado de manera significativa, permitiendo así almacenar, procesar y sintetizar señales de cualquier tipo.

El desarrollo de disciplinasde la matemática aplicada tan especializados como el aprendizaje máquina, data mining, learning to Rank para recuperación de información y el reconocimiento de patrones, han contribuido a la aparición de un amplio abanico de aplicaciones relacionadas con el tratamiento de datos y más enfocado al área que nos concierne, señales de audio. Un abanico que va desde el reconocimiento de voz, hasta la extracción señales distintas mezcladas en una sola como es el caso de la separación de melodías de distintos instrumentos en una misma canción.

En el caso concreto de este predictor, queremos recabar de manera muy sencilla información acerca de las tendencias musicales actuales, incluso de un futuro próximo. Por este motivo las aplicaciones que tienen cierto parecido a este predictor, intentan simular el proceso que sigue un individuo hasta averiguar si es una pieza musical es de su agrado o no, y lo más importante, descubrirlo cuantitativamente, pudiendo obtener el grado de satisfacción que evoca. Lo más interesante de todo esto es que se evalúa el gusto puro por la música, evitando ruido como el efecto de: las campañas de publicidad y la opinión sesgada del oyente hacia el artista.

Es casi ilusorio pensar que ahora las máquinas "opinen" y de alguna manera dicten los gustos musicales. Y es que va más allá de esto: la máquina predice los sentimientos que generan cierto tipo de canciones, pudiendo generar listas de reproducción según el estado de ánimo en el que te encuentres.

El prototipo cuyo desarrollo se explica en posteriores capítulos está enfocado a las tendencias musicales de Estados Unidos, pero es transportable a cualquier región del mundo (cambiando claro está, los datos de entrada y reajustando el modelo).

Este proyecto consta de dos fases bastante diferenciadas:

- Fase 1: La extracción de características de una pista de audio.

- Fase 2:La búsqueda del modelo que mejor se adapta a nuestro objetivo: la predicción del éxito.

1.1 MOTIVACIÓN

Poder predecir los gustos musicales, las canciones que serán un éxito o simplemente conocer las tendencias sonoras de un futuro inmediato es un estudio más que interesante.

Supone a nivel empresarial un ahorro de costes y disminución de riesgo por parte de las compañías discográficas y productoras, al apostar por músicos noveles. No sólo reduce la probabilidad de pérdidas económicas, sino que asegura beneficios.

A nivel artístico, es realmente cómodo saber qué características debe tener una canción para que ésta sea un éxito seguro. Esto no quita que la tarea de componer sea más fácil; dado que la restricción y la imposición de límites coartan la creatividad y la calidad de la innovación. Aun así tendría utilidad para el propio músico, no hay nada como comprobar si lo que se ha creado será de gusto generalizado.

Una posible finalidad de este método sería la de manipular la predisposición y preferencias de la multitud, aumentando así el consumo hacia ciertos artistas y o géneros. O mirándolo desde el lado positivo y pro activo, proporcionaría cultura, mejorando y potenciando el desarrollo del oído musical, dado que si se analiza el producto que se consume hoy en día, éste deja mucho que desear.

Sería más que curioso conocer cómo va variando el gusto musical a lo largo de los años.Utilizando el mismo modelo se podría averiguar. Simplemente habría que cambiar las muestras de entrada por canciones de las listas de mayores ventas de los últimos años y reajustar el prototipo.

1.2 OBJETIVOS

El objetivo principal de este proyecto es el desarrollo de un prototipo que prediga cuantitativamente el éxito de canciones de género Pop y potencialmente el puesto que va a ocupar en las listas Top 100.

Como objetivo secundario se trata de comprobar y verificar si este mismo prototipo sirve para canciones de otro género. Como ejemplo se capturan piezas de Jazz y Rock y se vuelcan en el análisis para comprobar si predice con la misma fidelidad. También averiguar qué características de la canción son las de mayor valor predictivo.

1.3 ORGANIZACIÓN DEL CONTENIDO

Capítulo 2. Revisión del estado del arte

Se detalla el entorno en el que se ha basado este estudio.Se divide en dos apartados principales:

2.1 Entorno Echonest: descripción de la herramienta desde donde se han obtenido las muestras utilizadas en este estudio.

2.2 Procesado de señal: explicación de la metodología que podría resolver el problema planteado en el proyecto.

Capítulo 3. Diseño e implementación

En este capítulo, se realiza una descripción detallada del desarrollo del prototipo final, distribuido en dos fases:

3.1 Obtención de muestras. Procedimiento realizado para conseguirlo y manipulación de los datos para su utilización en el software matemático Matlab.

3.2 Desarrollo de los modelos obtenidos con los datos de entrenamiento.

Capítulo 4. Valoración experimental

Se exponen una serie de gráficosque explican los resultados del uso de los modelos generados. Comparando sus errores cuadráticos, se elige el óptimo. Se observa el efecto de los patrones tanto en los datos de entrenamiento como en los de test.

Capítulo 5. Conclusiones

Con las conclusiones se resumen los resultados obtenidos con el diseño del predictor. Al mismo tiempo se proponen algunas ideas que podrían mejorar el prototipo, mejorando los resultados.

Junto a ellas se exponen futuros trabajos y líneas de investigación a seguir con el fin de justificar la elección de este estudio.

2

Revisión del estado del Arte

En este capítulo se hablará acerca de todos los recursos en los que este proyecto ha sustentado sus bases. Desde el servidor donde se han sacado las muestras de audio, hasta los métodos y algoritmos implementados para generar modelos ajustados para su posterior predicción.

Habrá una descripción general de cada uno de ellos y de manera más profunda, se entrará en detalles en aquellos aspectos más interesantes relacionados con el predictor de canciones.

2.1 ECHONEST[1]

Se trata de una compañía líder en "Music Intelligence" que proporciona el mayor repertorio dinámico de datos musicales en el mundo, alrededor de 30 millones de canciones y un trillón de datos. Se utiliza principalmente para ayudar a los desarrolladores de software a crear aplicaciones relacionadas con la música a través de su API.

Trabaja con un sinfín de partners importantes de la industria musical, como lo son: EMI, eMusic, MOG, MTV, Nokia, Rdio, Spotify entre otros.

El API[2] de Echonest permite invocar métodos que serán respondidos en JSON o XML directamente desde el navegador. Algunos de los métodos que a continuación vamos a explicar requieren un HTTP Post request Content-Type "multipart/form-data por lo que necesitaremos utilizar cURL.[3]

Json [4](Java Script Object Notation) es un formato texto utilizado para intercambio de datos. Está basado programación JavaScript. Sustituye a XML por su simplicidad a la hora de escribir y leer código. Es muy atractivo para los programadores por sus similitudes con los lenguajes: C,C++, C#, Java, JavaScript, Perl, Python, etc.

2.1.1 Métodos

A continuación se hará una breve descripción de los métodos que se pueden implementar con la herramienta API de Echonest. Dependiendo del tipo de información que se quiera obtener, hay diferentes algoritmos referentes a: artistas, canciones, catálogos, etc.

2.1.1.1 Métodos de la clase Artist

Devuelve un amplio rango de información y datos sobre cualquier artista.

Descripción de algunos métodos:

- *Familiarity* – Devuelve una estimación de cuan conocido es el artista alrededor del mundo.
- *Hotttnesss* – Devuelve un número que describe el éxito de un artista hoy en día.
- *Profile* – Obtiene información básica del artista:

[1] http://developer.echonest.com/
[2] http://developer.echonest.com/docs/v4
[3] cURL: es una herramienta para la transferencia de archivos con sintaxis URL mediante línea de comandos. Soporta certificación SSL, HTTP POST,HTTP PUT, FTP.
[4]http://www.json.org/

- Nombre
- EchonestID: el identificador en Echonest del artista.
- MusicBrainzIDGet: el identificador del artista en Music Brainz (partner de Echonest)

- *Search* – Busca al artista seleccionado con los siguientes parámetros posibles.

2.1.1.2 Métodos de la clase Song

Funciones para obtener información y datos sobre canciones. También permite descripción de parámetros *bucket*, algunos de ellos pueden ser:

- Search – Búsqueda de canciones pudiendo pasar como criterio de búsqueda cualquier cosa que se pueda imaginar: por estilo, artista, estado emocional, tempo, duración, energía, éxito, etc.

Por ejemplo: suponer que se quiere buscar por las siguientes características: bailables y rápidas canciones de rock.

http://developer.echonest.com/api/v4/song/search?api_key=FILDTEOIK2HBORODV&style=rock&min_danceability=0.65&min_tempo=140&results=2

Cuya respuesta sería en formato Json:

```
{
    "response": {
        "songs": [

// Devuelve la lista de canciones como resultado la búsqueda por los parámetros descritos.
            {
                "artist_id": "AR136921187FB4B8D2",

// Identificador de artista.
                "artist_name": "McQueen",

// Nombre del artista
                "id": "SOFBJYM130516DF5F6",

// Identificador de la canción
"title": "Running Out of Things to Say (video)"

//Título de la canción.
            },
            {
```

```
                "artist_id": "AR136921187FB4B8D2",
"artist_name": "McQueen",
                "id": "SOZRITE12DA5949F08",
                "title": "Running Out of Things to Say (video)"
}
        ],
        "status": {
```


// Describe el estatus de la respuesta, si el código es cero entonces es exitosa.


```
"code": 0,
            "message": "Success",
            "version": "4.2"
        }
    }
}
```


- Profile – Obtener información de las canciones dadas por medio de su songID.

2.1.1.3 Métodos de la clase Pista

Métodos para analizar y obtener información sobre cualquier pista.

- Analysis – resumen sobre el pista, descripción de propiedades sonoras como: tempo, key signature, time signature, mode y loudness, junto con información detalladas de la estructura de la canción (sections), estructura del ritmo (bars,beat tatums) e información detallada sobre el timbre, pitch y la envolvente sonora.

- Profile

- Upload– permite subir una pista al servidor de Echonest para analizarlo. Este método requiere un parámetro url donde encontrar la canción o un archivo de audio local. Se utiliza cuando no se encuentra la canción en Echonest, aunque da bastantes errores y es recomendable emplearlo como última opción. Es necesario el uso de Curl.

Ejemplo:

```
curl -F "api_key=FILDTEOIK2HBORODV" -F "filetype=mp3" -F
"pista=@audio.mp3"
http://developer.echonest.com/api/v4/pista/upload
```

2.1.1.4 Métodos de la clase Catalog

Permite el manejo de catálogos (colección de ítems musicales (artists, songs) que pueden ser usados como entradas de parámetro para métodos del API).

- *Create*- genera un catálogo.Puede ser de dos tipos, según la información que contendrádicho catálogo.
  -Artista
  -Pista

Ejemplo:

Escribimos en DOS por línea de comandos el siguiente código donde se realiza una petición de creación de un catálogo al servidor de Echonest, asociado a una cuenta de cliente "api_key", definiendo el formato (json), el tipo (artista) y el nombre de éste (test_artist_catalog).

```
curl -F "api_key=FILDTEOIK2HBORODV" -F "format=json" -F
"type=artist" -F "name=test_artist_catalog"
"http://developer.echonest.com/api/v4/catalog/create"
```

Como respuesta, devuelve el estatus de la petición que en este caso fue exitosa, la identificación del catálogo en las bases de datos del servidor (ID) y también el tipo (artista).

```
{
"response":{
"status":{
"code":0,
"message":"Success",
"version":"4.2"
},
"name":"test_artist_catalog",
"id":"CAOFUDS12BB066268E"
"type":"artist",
}
}
```

- *Update* - Actualiza (añadiendo/eliminando) ítems del catálogo.

La respuesta al proceso ocurre de forma asíncrona a la llamada, por lo que el método update devuelve un "ticket" que puede ser usado para comprobar el estatus de la actualización.

Ejemplo:

Se escribe en DOS por línea de comandos el siguiente código donde se realiza una petición de actualización de un catálogo al servidor de Echonest, asociado a una cuenta de cliente "api_key", definiendo el formato de los datos que se van a pasar por parámetro (json), el formato del catálogo (json), el ID del catálogo y se invoca al archivo de formato .json que contiene toda la información de la actualización.

```
curl -X POST
"http://developer.echonest.com/api/v4/catalog/update" -F
"api_key=FILDTEOIK2HBORODV" -F "data_type=json" -F "format=json"
-F "id=CAJTFEO131216286ED" -F "data=@data_file.json"
```

Como respuesta da una muy parecida a la del método anterior, salvo que devuelve un código llamado ticket.

```
{
"response": {
    "status": {
      "code": 0,
      "message": "Success",
"version": "4.2"
    },
    "ticket": "d298131d5f189c73bd9a8ff706621443"
  }
}
```

- Read - Devuelve los datos almacenados en el catálogo.

Por ejemplo:

Se le pide al servidor por la línea de buscador del navegador que lea un catálogo con un ID determinado, y además que incluya en la salida un resumen de las características de audio de cada ítem del catálogo

http://developer.echonest.com/api/v4/catalog/read?api_key=FILDTEOIK2HBORODV&format=json&id=CAJTFEO131216286ED&bucket=audio_summary

Responde con el estatus de la petición junto con información relevante de cada ítem/canción, como lo es: el nombre, su id_song, el nombre del artista, el identificador del artista, el identificador del ítem, la fecha en el que se añadió al catálogo, el número de veces que se ha reproducido esa canción y por último el resumen pedido.

```
{
"response":{
"status":{
"code":0,
"message":"Success",
"version":"4.2"
},

"catalog":{
"name":"test_song_catalog",
"items":[
{
"song_id":"SOYRVMR12AF729F8DC",
"song_name":"Harmonice Mundi II",
"request":{
"item_id":"0CF07A178DBF78F7",
"song_name":"Harmonice Mundi II",
"artist_name":"Six Organs OfAdmittance"
},
"artist_name":"Six Organs of Admittance",
"play_count":4
"date_added":"2010-10-15T15:18:32",
"artist_id":"ARK3D5J1187B9BA0B8",
"foreign_id":
"CAGPXKK12BB06F9DE9:song:SOYRVMR12AF729F8DC",
"audio_summary":{
"key":4,
"analysis_url":"https://echonest-
analysis.s3.amazonaws.com:443/TR/TRDJDSS1264E5BB481/3/full.json?
Signature=dQg3hdA7MgJ704cid6kctP7IhW8%3D&Expires=1287159130&AWSA
ccessKeyId=AKIAIAFEHLM3KJ2XMHRA",
"energy":0.0032362399065479839,
"tempo":113.429,
"mode":1,
"time_signature":4,
"duration":178.85995,
"loudness":-23.824000000000002,
"danceability":0.16339223882299694
}
"artist_name":"Wilco",
"pista_id":"TRMAZTJ123E85978B7",
"play_count":10,
"date_added":"2010-10-15T15:24:17",
"foreign_id":"CAGPXKK12BB06F9DE9:song:2EC046EA14A873F8",
"artist_id":"AR6SPRZ1187FB4958B",
"audio_summary":{
"key":2,
"analysis_url":"https://echonest-
analysis.s3.amazonaws.com:443/TR/TRMAZTJ123E85978B7/3/full.json?
Signature=TjzpSgav0iNDDanceabilityCNvG1OzYd0QjE%3D&Expires=12871
59130&AWSAccessKeyId=AKIAIAFEHLM3KJ2XMHRA",
"energy":null,
"tempo":111.544,
"mode":0,
"time_signature":4,
```

```
"duration":218.85342,
"loudness":-10.044,
"danceability":null
}
}
],
"start":0,
"total":2,
"type":"song",
"id":"CAGPXKK12BB06F9DE9"
}
}
}
```


- Delete - Elimina el catálogo entero.

- List - Devuelve una lista con todos los catálogos creados con un determinado ApiKey[5].

Algunos métodos como: search, similar, top_hottt y profile aceptan un parámetro llamado *bucket*. Esta variable permite especificar qué otros datos adicionales pueden ser devueltos con cada artista. Múltiples buckets pueden ser definidos de manera simultánea permitiéndote enviar una solicitud múltiple de datos en una simple llamada.

2.1.2 Análisis sonoro

El analizador de Echonest "Analyze" es una herramienta de análisis de audio disponible para los usuarios. El programa funciona de la siguiente manera: captura el archivo de audio digital del disco y genera un texto de formato JSON que describe la estructura de la pista y el contenido musical, incluyendo entre otros: ritmo, timbre y frecuencia fundamental.

Utiliza técnicas de *machine listening* para simular la percepción de la música en las personas. Incorpora principios de la psicoacústica, percepción musical, y aprendizaje adaptativo para modelar el proceso tanto cognitivo como físico de la escucha.

[5] ApiKey: código o clave de usuario.

La salida contiene una descripción completa de todos los eventos, estructuras y atributos globales de la canción, como por ejemplo: key, loudness, time signature, tempo, beats, sections, armonía.

Permite a los desarrolladores crear aplicaciones relacionadas con la manera de la gente de escuchar e interactuar con la música. Gracias a esto se puede:

- Interpretar: entender, describir y representar música.
- Sincronizar: música con otros sonidos, vídeos, textos y otros contenidos multimedia.
  Para por ejemplo la creación automática de bandas sonoras.
- Manipular: remix, mezclas, o procesar la música mediante transformación de su contenido.

2.1.2.1 Información devuelta por *Analyse*. Atributos acústicos.

Un atributo acústico es una estimación de la calidad de la canción. Está modelado mediante aprendizaje automático y es devuelto normalmente como un número entre el rango 0.0-1.0, en otros casos >1. Se verá más adelante cuales son los casos.

Estos atributos pueden ser usados como filtros de búsqueda, entre otras cosas.Algunos de estos atributos disponibles son los explicados a continuación.

- *Tonalidad*

Define las distancias entre notas y la tónica o centro tonal en función de la consonancia sonora.La tonalidad proporciona la estructura de movimiento alrededor del sistema armónico mediante acordes y escalas asociados para realizar progresiones musicales que siempre suenen bien (en consonancia). Cada nota o acorde de la tonalidad recibe un grado musical, según la posición que ocupa la nota en la escala diatónica.

- I (primer grado): tónica
- II (segundo grado): supertónica
- III (tercer grado): mediante
- IV (cuarto grado): subdominante
- V (quinto grado): dominante

- VI (sexto grado): superdominante o submediante
- VII (séptimo grado): sensible (en la escala diatónica mayor) o subtónica (en la escala diatónica menor)

- *Danceability*

Diseñado para describir cuantitativamente la adecuación de una canción para ser bailada. Cuanto más adecuada sea, más próximo es su valor a la unidad. Es una combinación de elementos acústicos incluidos el tempo, uniformidad rítmica, la fuerza y sobretodo la regularidad.

- *Energía*

Representa una medida perceptual de la intensidad y la actividad que deja la canción. Una canción enérgica es rápida y ruidosa por lo general. Características perceptuales que contribuyen a la medida de la energía incluye: rango dinámico[6], volumen percibido, timbre[7] y entropía, siendo la entropía la cantidad de "ruido" o "desorden" que contiene un sistema.

- *Speechiness*

Detecta la presencia de palabras habladas en la pista. También tiene una valoración del cero al uno. Las pistas con mayor contenido hablado, (como por ejemplo un audio libro, poesía...) más cerca está su valor del uno. Valores por encima de 0.66 describen pistas que están hechas en su totalidad de voz hablada. Valores entre 0.33 y 0.66 describen pistas que contienen música y voz (ejemplo: rap). Por último, cuando el valor está por debajo de 0.33 son canciones completamente instrumentales.

- *Modo*

Indica la modalidad (mayor o menor) de la canción. Es el tipo de escala de donde se deriva el contenido melódico.

[6]El margen que hay entre el *nivel de referencia* y el ruido de fondo de un determinado sistema, medido en decibelios.
[7]El matiz característico de un sonido, que puede ser agudo o grave según la altura de la nota que corresponde a su resonador predominante.

Las escalas musicales en modo mayor son las que tienen una distancia tercera mayor entre el primer y el tercer grado, y una tercera menor entre el tercer y el quinto grado. A continuación se enseña un ejemplo de la secuencia de los intervalos[8] en la escala Do Mayor.

	T		T		S		T		T		T		S	
DO		RE		MI		FA		SOL		LA		SI		DO

Donde T es tono y S semitono. Es decir, la distancia entre DO y Re es de un tono, mientras que de Mi a Fa es de medio tono.

En cambio, con el modo menor la distancia entre su primer y tercer grados es de tercera menor (un tono y medio). Así pues, la secuencia de los intervalos en el ejemplo a continuación es:

Escala de La Menor:

	T		S		T		T		S		T		T	
LA		SI		DO		RE		MI		FA		SOL		LA

- *Tempo*

Devuelve el tempo de una pista en pulsos por minuto (BPM)[9]. El tempo es la velocidad de una pieza musical. También se puede definir como la duración en tiempo de la figura fundamental de la pieza (generalmente es la negra).

Es un factor tremendamente importante dado que determina el estado de emoción de la canción.

- *Sonoridad*

Formalmente se define como "Es el atributo que permite al oído humano ordenar un sonido por su intensidad, en una escala que va desde lo silencioso hasta lo ruidoso".

Es una medida que describe la manera en que el ser humano percibe la intensidad de un sonido. De cualquier manera este parámetro también está bajo la influencia de características físicas, como lo son: la presión sonora, la banda de frecuencias en las que es audible y la duración.

[8] Distancia entre dos notas.
[9] Medida de unidad de tiempo en la música.

La siguiente gráfica muestra la relación (dB) existente entre la frecuencia y la intensidad de dos sonidos para que sean percibidos igual de fuertes.

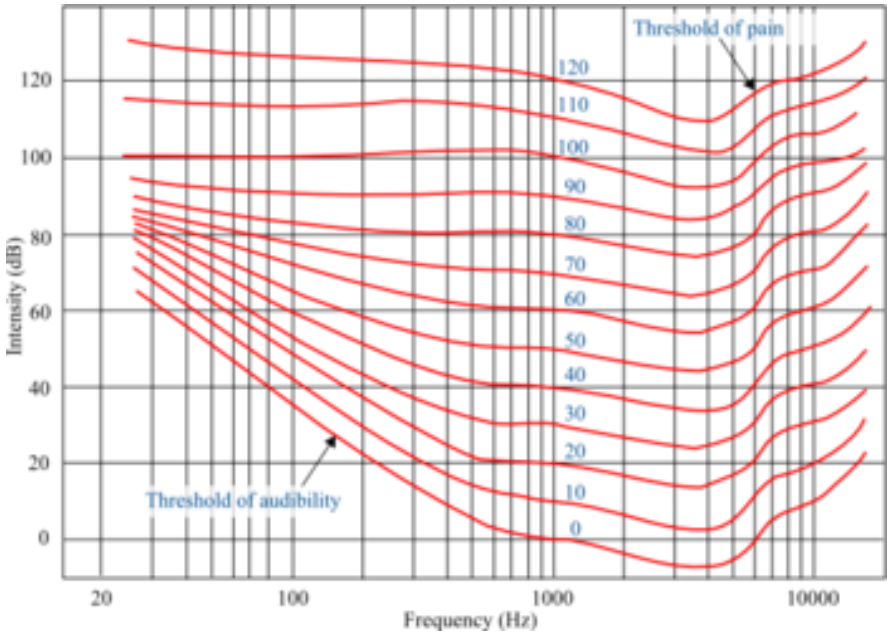


Figura 1. Curvas Isofónicas[10]

- *Duración*

Hace referencia a la duración de la pista en segundos. Es una medida precisa, y no estimada, dado que es obtenida por el decodificador.

[10] http://es.wikipedia.org/wiki/Sonoridad_(sicoac%C3%BAstica)

2.2 REGRESION LINEAL

El análisis de regresión lineal es una técnica estadística utilizada para estudiar la relación entre variables: una variable dependiente de salida Y, variables independientes(a priori) Xi y término aleatorio bajo la presunción de normalidad.

Si la relación es de dos variables se llama regresión lineal simple,

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

Si es de más de dos: regresión lineal múltiple,

$$Y_i = \beta_0 + \sum \beta_i X_{ip} + \varepsilon_i$$

Es una herramienta muy versátil, se adapta a una gran variedad de posibilidades. En nuestro caso lo utilizaremos como uno de los medios de predicción del éxito de una canción.

Modelo Matricial

$$Y = X\beta + \varepsilon$$

Notación:

$$Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \quad X = \begin{pmatrix} 1 & x_{11} & x_{21} & \cdots & x_{k1} \\ 1 & x_{12} & x_{22} & \cdots & x_{k2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{1n} & x_{2n} & \cdots & x_{kn} \end{pmatrix} = \left(\vec{1}, \vec{X}_1, \vec{X}_2, \ldots, \vec{X}_k\right) \quad \beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \end{pmatrix}$$

X Es la variable explicativa, el regresor o la variable independiente.

Y Es la variable observada, dependiente o respuesta.

β Es el vector de parámetro o el vector de pesos.

ε Es el vector de variables aleatorias representando el error en la relación entre x e y. De distribución normal de media cero y varianza v. Independencia entre sus variables.

$$\varepsilon \rightarrow N(0,\sigma^2)$$

En la regresión lineal los datos son modelados usando funciones de predicción lineal, los parámetros del modelo son estimados según datos de entrada. Este tipo de modelos se llaman modelos lineales pues asumen que la esperanza de Y dado X es lineal.

$$E(Y \mid X) = \beta_0 + \sum_{i=1}^{N} \beta_i X_i = f(X)$$

Ahora, β y **ε** son desconocidos en la ecuación de regresión. De hecho **ε** es difícil de descubrir por el hecho de que varía para cada observación y. β en cambio, se puede obtener de manera aproximada examinando todas las posibilidades de combinación entre Y y X.

Existen muchos métodos que explicaremos a continuación para la estimación de β. Una vez obtenido el vector de coeficientes de regresión, el valor estimado de la salida (Y) puede definirse como:

$$Y_i = \sum \hat{\beta}_k X_{ki} + \hat{\varepsilon}_i$$

Ventajas de los modelos de regresión lineal

Los modelos lineales son ventajosas por:

- Aunque son herramientas clásicas son muy útiles.
- Simplicidad.
- Permite una fácil interpretación del efecto de los regresores sobre la respuesta Y.

2.2.1 Interpolación y ajuste

Para saber qué modelo se debe utilizar para predecir el comportamiento de una variable dependiente de otras, lo primero de todo es realizar una gráfica de dispersión, que muestra la relación entre estas.

En el ejemplo se expone un diagrama de dispersión entre muestras de peso y talla.

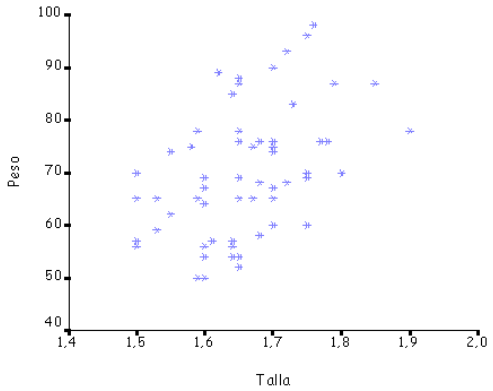


Figura 2. Ejemplo de diagrama de dispersión [11]

Hay tres posibles formas de distribución de los puntos:

- La nube de puntos forma parte total de una función matemática, luego existe una dependencia funcional entre las dos variables, se puede definir y en función de x.

- La nube de puntos no coincide exactamente con la figura de una función matemática pero se aproximan a ella con mayor o menor medida, luego existe una dependencia estadística aproximada entre las dos variables.

- La nube de puntos no se concentra en ninguna función matemática definible, sino se distribuyen en una región del plano, las variables son independientes entre sí. No existe ninguna relación.

El problema de ajuste se puede resolver por numerosas vías, destacamos tres de ellas:

- *Método de los mínimos cuadrados.*
- *Método robusto.*
- *Método Stepwise.*

[11] http://www.fisterra.com/mbe/investiga/graficos/graficos.asp

2.2.1.1 Mínimos cuadrados

Por referencia se define como un procedimiento matemático de búsqueda de la mejor curva ajustada dado una serie de puntos por la minimización de la suma de los cuadrados de los residuos.

El residuo es la diferencia de distancia entre el punto y la curva modelo. Se utiliza la suma de cuadrados de los residuos en vez de la suma absoluta debido a que permite que los residuos sean tratados como cantidades continuas diferenciales. Los puntos que no se ajustan en la curva pueden inferir un efecto desproporcionado en el ajuste, esta es una propiedad que puede o no ser deseada según el problema propuesto.

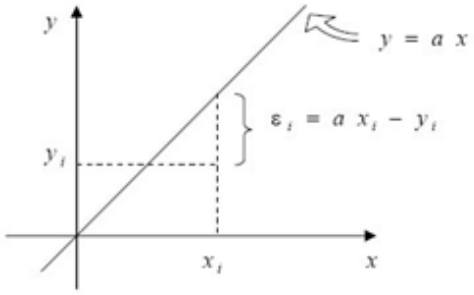


Figura 3. Error de predicción[12]

En cuanto a su interpretación geométrica, se puede decir que $\hat{y}$ es la proyección ortogonal de y sobre el plano, siendo el error la distancia entre los vectores y e $\hat{y}$.

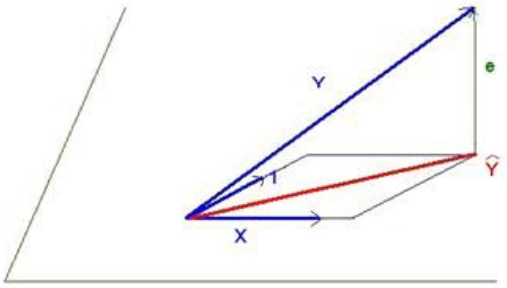


Figura 4. Interpretación geométrica del residuo

[12] http://elhl.blogspot.com.es/2008/07/metodo-de-minimos-cuadrados.html

El objetivo consiste en ajustar los parámetros de un modelo de la mejor manera al conjunto de datos. Este modelo tiene la forma $f(x, \beta)$, donde los parámetros ajustables están contenidos en el vector vector β. Un residuo por definición es la diferencia entre el valor actual de la variable dependiente y el valor predicho por el modelo.

$$\boldsymbol{\varepsilon} = \mathbf{y} - \mathbf{X} \cdot \boldsymbol{\beta}$$

El vector óptimo encontrado por el método de los mínimos cuadrados es aquel que minimice la suma de los residuos al cuadrado.

$$SCE = \sum_{i=1}^{n} \varepsilon_i^2 = \boldsymbol{\varepsilon}'\boldsymbol{\varepsilon} = (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})'(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})$$

Si desarrollamos el anterior resultado nos queda:

$$\mathbf{X}'\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}'\mathbf{y}$$

Y despejándose la variable paramétrica

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$$

Se verifica que esta estimación es el estimador de mínimo error cuadrático medio de β.

El ajuste lineal por mínimo cuadrados es el procedimiento más simple y común aplicado a la regresión lineal. De hecho, si la relación entre dos valores graficada es conocida sin añadir o multiplicar constantes, es muy común transformar los datos para que la curva resultante sea una recta. Se aplica de igual manera a funciones exponenciales, logarítmicas y polinómicas.

Para ajuste no lineal por mínimos cuadrados de un número indeterminado de parámetros, este método puede ser aplicado iterativamente, una vez sea linearizada la función y comprobada su convergencia.

2.2.1.2 Ajuste por regresión robusta

Es un modelo diseñado para saltar limitaciones de los tradicionales métodos paramétricos. Dichos procedimientos funcionan perfectamente para datos que se ciñen al modelo pero no

sucede lo mismo para aquellos que no lo están. El modelo robusto no se deja afectar mucho por lo outliers[13]. Los outliers son observaciones que no siguen el patrón de otras observaciones, los datos atípicos. No pasaría nada si estos datos fueran observaciones límite de una distribución normal, pero si resultan ser un dato que viola algunos de los supuestos de los mínimos cuadrados entonces comprometerá el resultado de la regresión.

Este modelo es menos sensible a errores muy grandes en pequeñas regiones que los modelos corrientes, como el de mínimos cuadrados.

El ajuste por robust regression consiste en asignarle un peso a cada observable. Esto se hace de manera automática e iterativa mediante un proceso llamado IRLS.

$$\boldsymbol{\beta}^{(t+1)} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} w_i(\boldsymbol{\beta}^{(t)}) \left| y_i - f_i(\boldsymbol{\beta}) \right|^2.$$

Su objetivo es encontrar el M-estimador como una forma de mitigar la influencia de los datos atípicos en una distribución distinta a la normal.En la primera iteración se le asigna igual peso a todas las observaciones y los coeficientes del modelo se estiman aplicando mínimos cuadrados. Se otorgan los pesos de tal manera que se da menos importancia a los puntos más alejados al modelo de predicción de la iteración anterior. A continuación se recalcula los coeficientes, con el mismo procedimiento que la primera iteración.

El algoritmo se detiene cuando los valores de los coeficientes convergen con una tolerancia específica.

Algoritmo IRLS

Para encontrar los parámetros $\beta = (\beta_1, \ldots, \beta_k)^{\mathrm{T}}$ que minimizan la norma Lp en un problema de regresión lineal,

$$\arg\min_{\boldsymbol{\beta}} \left\| \mathbf{y} - X\boldsymbol{\beta} \right\|_p = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} \left| y_i - X_i\boldsymbol{\beta} \right|^p,$$

El método IRLS resuelve el problema de mínimos cuadrados en la iteación (t+1):

$$\boldsymbol{\beta}^{(t+1)} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} w_i^{(t)} \left| y_i - X_i\boldsymbol{\beta} \right|^2 = (X^{\mathrm{T}} W^{(t)} X)^{-1} X^{\mathrm{T}} W^{(t)} \mathbf{y},$$

[13]Valor que se aleja mucho de la nube de puntos.

Donde $W^{(t)}$ es la matriz diagonal de los pesos, cada elemento de esa matriz tiene la siguiente forma:

$$w_i^{(t)} = \left| y_i - X_i \boldsymbol{\beta}^{(t)} \right|^{p-2}.$$

2.2.1.2.1 Regresión robusta vs mínimos cuadrados

Se muestra un ejemplo comparativo de los dos métodos ajustándose a una línea recta de forma: y = 10-2*x.

Con el mismo conjunto de datos de entrada, se obtienen los siguientes resultados:

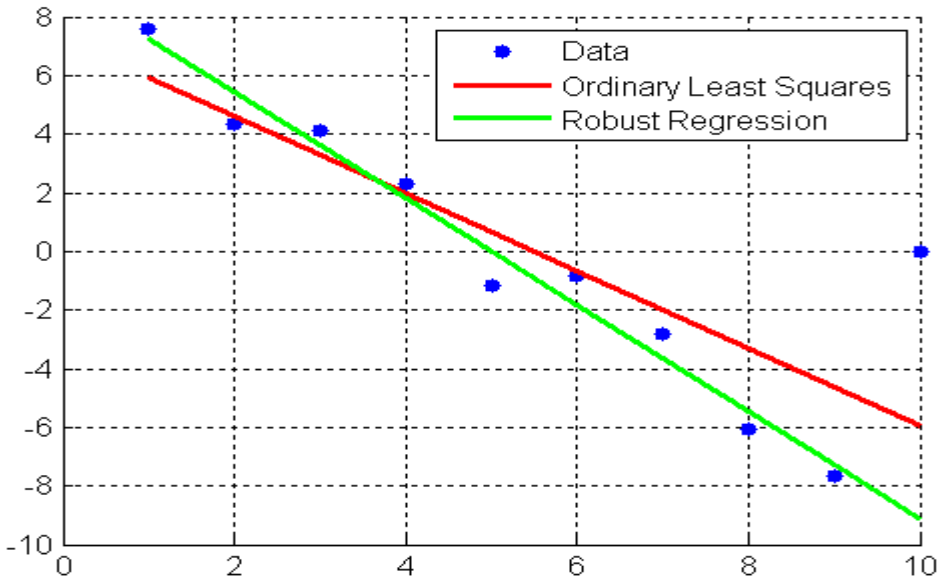


Figura 5. Regresión robusta vs Lineal

Se puede observar la influencia que ejerce un dato atípico y aislado sobre el ajuste por mínimos cuadrados. En cambio con el robust regression apenas es sensible.

2.2.1.3 Selección de variables. Regresión Stepwise

Intuitivamente, cuando se tiene un conjunto de muchas variables es lógico preguntarse si todas ellas deben ser parte del modelo de regresión o no y más importante, cuáles son las que deben permanecer.

La selección de variables es un procedimiento estadístico de gran importancia dado que no todos los predictores tienen la misma importancia. Incluso si unas son combinación lineal (correladas) de otras pierden todo su valor informativo, perjudicando la calidad del modelo.

Conviene trabajar con un número de variables predictivas bajo pues la carga computacional es mucho menor y por ello el costo ("Principio de Parsimonia")

El objetivo principal de los métodos que mencionaremos a continuación es conseguir que el modelo elegido sea "deseable", de manera secuencial siguiendo ciertos criterios. Bajo la hipótesis de que la variable respuesta *y* tiene una relación lineal con sus predictores, se crea un nuevo método "paso a paso". Es una técnica que envuelve:

- Identificación del modelo inicial.
- "*Stepping*" iterativo: alteración del modelo en el paso previo de manera repetitiva, añadiendo o eliminando una variable predictiva acorde al criterio *stepping*.
- Condición de parada: se termina la búsqueda cuando *stepping* no es posible cuando no cumple un criterio o cuando se especifica por parámetro el número máximo de iteraciones que se quieren.

BACKWARD ELIMINATION

Se empieza con el modelo completo, con todas las variables predictivas. En los siguientes pasos se va eliminando una de ellas. La manera de saber cuál se elimina es la siguiente:

- "Aquella variable que tiene el estadístico de `t`, en valor absoluto, más pequeño entre las variables aún en el modelo"
- "Aquella variable que produce la menor disminución en el residuo al cuadrado al ser eliminada del modelo"

Estos pasos se vuelven a repetir hasta que se llegue a la condición de parada o cuando se llegue al número de variables `p` deseado.

FORWARD SELECTION

Comienza con la variable predictiva con mayor correlación con la variable respuesta y. A continuación se añade aquella variable que cumpla con la siguiente condición. *"Aquella variable que produce el mayor incremento en el residuo al cuadrado al ser añadida al modelo"*

Se termina la operación de la misma manera que con el Backward elimination.

STEPWISE SELECTION

Es la combinación de los procesos anteriores. Comienza igual que el Fordward Selection pero en cada paso se coteja si cada una de las variables incluidas en el modelo puede ser eliminada (Backward Elimination).

La lógica subyacente de esta recursión consiste en conservar las variables independientes que contienen información relevante, prescindiendo de aquellas que sean redundantes a las anteriores. En realidad, este procedimiento es más útil para conocer las variables más influyentes que en construir modelos de predicción.

Estas últimas técnicas de regresión se pueden unificar con la siguiente fórmula:

[14]

$$\hat{\beta} = \arg\min_{\beta} \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|^q \right\}$$

λ=0: mínimos cuadrados.

λ>0, q=0: stepwise selection.

λ>0, q=1: Lasso.

λ>0, q=2: Ridge regression

2.2.1.4 Métodos de Regularización

[14]http://www.ccee.edu.uy/jacad/2012/x%20area%20y%20mesa/METODOS%20CUANTITATIVOS/Mesa%203/3-Analisis%20de%20datos%20en%20grandes%20dimensiones.%20Estimacion%20y%20seleccion%20de%20variables%20en%20regresion..pdf

Utilizan penalizaciones para reducir el número de coeficientes de regresión.

Ejemplos de este procedimiento:

-Ridge Regression: minimiza el método de mínimos cuadrados de la siguiente manera:

$$\sum_{j=1}^{p} \beta_j^2 \leq s$$

-LASSO: emplea otro método de optimización de los mínimos cuadrados:

$$\sum_{j=1}^{p} |\beta_j| \leq s$$

2.2.1.4.1 Ridge Regression

Esta técnica nace en el 70 para luchar contra la colinealidad, problema generado por un modelo lineal estimado por mínimos cuadrados. La colinealidad es la fuerte correlación que existe entre variables de un modelo → esto provoca que la matriz XX^t tenga determinante cero, es decir sea singular y por lo tanto no invertible. Si esto sucede, no podemos estimar los parámetros del modelo, sino una relación lineal entre ellos.

El efecto de la colinealidad es el aumento de las varianzas de los estimadores de los coeficientes de regresión, si sumamos a esto que la matriz inversa de XX^t tiende a ∞

$$|X'X| \simeq 0 \longleftrightarrow \frac{1}{|X'X|} \approx \infty$$

Y las varianzas de las perturbaciones:

$$Cov(\beta) = \sigma_\epsilon^2 (X'X)^{-1} = \sigma_\epsilon^2 \frac{1}{|X'X|} adj(X'X)'$$

Al estimar los coeficientes de la regresión estos van a salir poco precisos y tremendamente sensibles a las muestras si usamos el método de mínimos cuadrados para la estimación.

Al igual que la regresión lineal simple, la regresión de arista buscar la solución a este simple problema: $A\mathbf{x} = \mathbf{b}$, sin embargo cuando este no está bien planteado, se reemplaza por el estándar: mínimos cuadrados, que como vimos con anterioridad minimizan $\|A\mathbf{x} - \mathbf{b}\|^2$ (norma euclídea). La condición es que A debe ser singular, lo que no mejora el primer planteamiento.

La regularización, que mejora el condicionamiento del problema, añade el término de regularización a esta norma: $\|A\mathbf{x} - \mathbf{b}\|^2 + \|\Gamma\mathbf{x}\|^2$. y devuelve una solución numérica de la siguiente notación matricial:

$$\hat{\boldsymbol{\beta}}^{ridge} = (X^T X + \lambda I)^{-1} X^T \mathbf{y}$$

Siendo el parámetro de ridge regression.Así pues el problema del método se convierte en averiguar el valor de λ óptimo.

El valor λ determina,

- El tamaño de los coeficientes de regresión.
- La cantidad de regularización.

Las posibilidades de solución empleando este procedimiento son infinitas, existen tantas como parámetros de regresión.

En la siguiente ilustración se enseña los posibles coeficientes de regresión (β) según λ. Se puede elegir el valor de λ óptimo simplemente con analizar esta gráfica. Se escoge aquel que sea el menor de todos los considerados para los cuales se estabilice los coeficientes estimados.

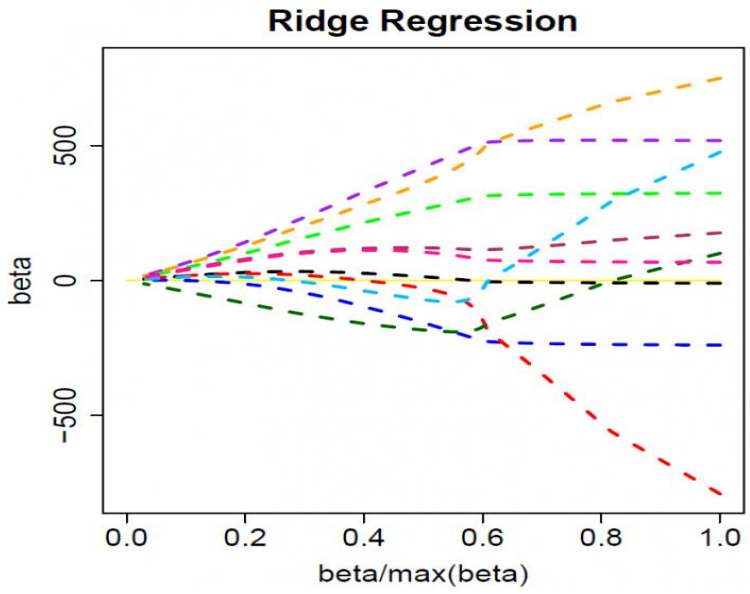


Figura 6. Trazado de un modelo dado λ

Si $\lambda \rightarrow 0$ Se obtiene la solución por mínimos cuadrados.

Si $\lambda \rightarrow \infty$ El valor de los coeficientes de regularización serán nulos: $\hat{\beta}_{\text{ridge } \lambda=\infty} = 0$

Para elegir un correcto valor de λ se debe tener en cuenta que:

- Es importante escoger un valor de λ para el que los coeficientes no varíen rápidamente.
- Para valores pequeños y positivos se reduce la varianza de los estimadores, pero en ocasiones provoca un error cuadrático medio más pequeño que empleando estimación por mínimos cuadrados.

Ridge regression se puede ver como un caso particular de los casos de regularización restringiendo la norma L_2 de los coeficientes del modelo, pudiéndose obtener β ridge como:

$$\hat{\beta}^{ridge} = \arg\min_{\beta_0,\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\}$$

Ridge regression es también un estimador lineal $(\hat{\mathbf{y}} = \mathbf{H}\mathbf{y})$, con

$$\mathbf{H}_{\text{ridge}} = \mathbf{X}(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T$$

NOTA: El uso de este tipo de penalización hace que la estimación de y sea no lineal.

Las técnicas de regularización se han empleado para evitar sobreajuste [15]. Añadiendo información al modelo los mecanismos de regularización lo convierten en más exacto. Funciona aplicando penalizaciones cuanto más complejo sea el modelo. De esta manera reduce el número de coeficientes de regresión.

En la figura 7secomparalos contornos de la función de error de los métodos: ridge y modelo lineal. Para ridge la curva cumple la siguiente ecuación $\beta 1^2 + \beta 2^2 \leq t^2$.

Como se puede observar el coeficiente de regresión estimado óptimo es aquel tangente a la elipse.

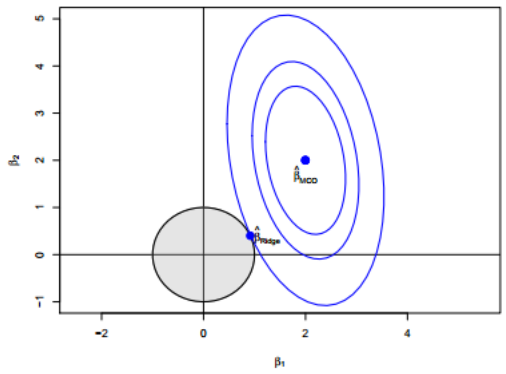


Figura 7. Descripción gráfica de la estimación Ridge en dos dimensiones [16]

2.2.1.4.2 Lasso

Es otro método regresión lineal regularizada muy parecido al ridge regression pero con la diferencia en la penalización. A partir de cierto valor del parámetro de complejidad la estimación de los coeficientes de regresión es nula y en otros no, por lo tanto realiza una especie de selección de variables. Con Ridge Regression todos los coeficientes son forzados a tender a cero sin llegar a serlo.

---

[15] Efecto de sobreentrenar un algoritmo de aprendizaje, prediciendo bien los datos de entrenamiento, pero no los de test.

[16] http://www.ccee.edu.uy/jacad/2012/x%20area%20y%20mesa/METODOS%20CUANTITATIVOS/Mesa%203/3-Analisis%20de%20datos%20en%20grandes%20dimensiones.%20Estimacion%20y%20seleccion%20de%20variables%20en%20regresion..pdf

Luego el objetivo de LASSO no es solo lograr la estabilización de las estimaciones, que también lo pretende Ridge, sino también generar modelos más estables mediante la selección de variables.

$$\hat{\beta}^{LASSO} = \arg\min_{\beta_0,\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}$$

Al tiempo que λ crece, el número de coeficientes β distintos de cero decrece. A diferencia de Ridge Regression utiliza penalización de norma L_1 de B.

En la siguiente gráfica se puede observar que los parámetros en LASSO cortan en cero, mientras que en Ridge no:

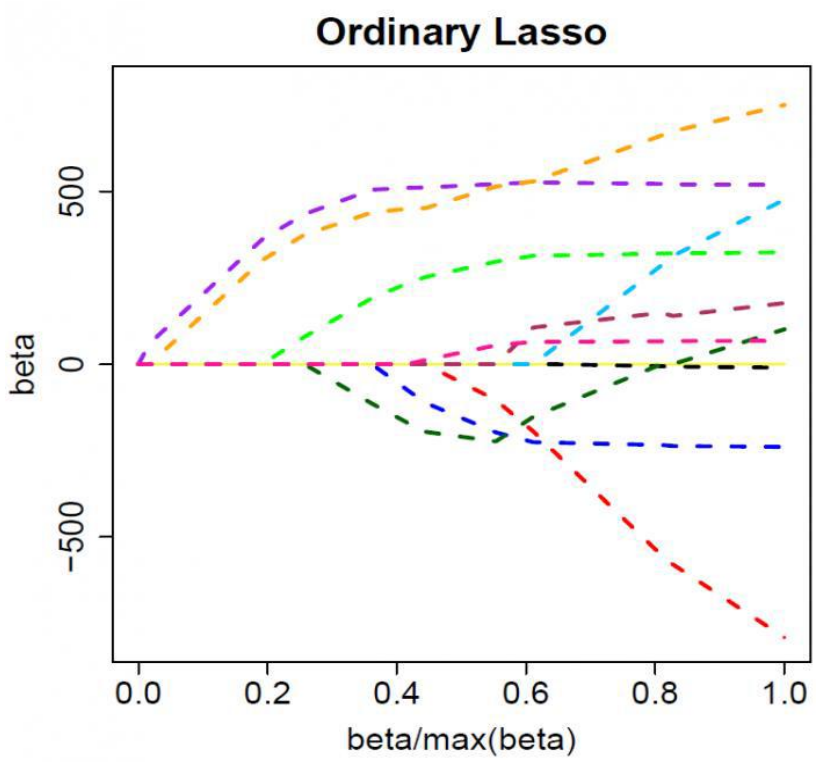


Figura 8. Gráfica de coeficientes de regresión

LASSO es una buena opción cuando tenemos muchas variables en el modelo a estimar.

En la figura 9 se compara los contornos de la función de error de los métodos: lasso y modelo lineal. Para lasso la curva cumple la siguiente ecuación : $|\beta 1| + |\beta 2| \leq t$ .Como se puede

observar el coeficiente de regresión estimado óptimo es aquel donde se anula alguna de las dos variables beta y corta con el contorno elíptico de la función de error de mínimos cuadrados.

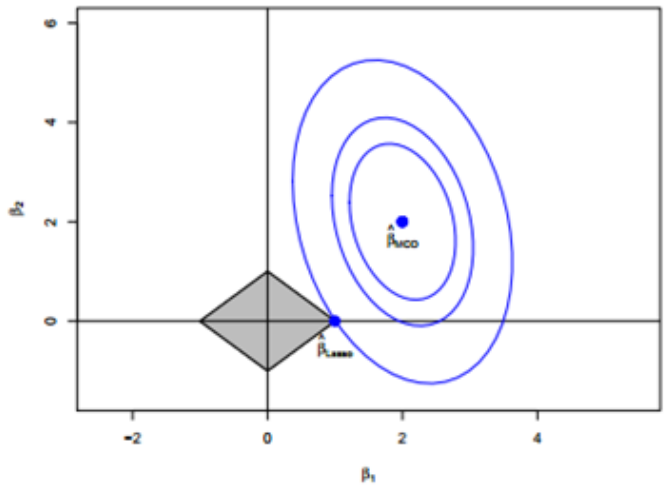


Figura 9. Descripción gráfica de la estimación Lasso en dos dimensiones[17]

2.3 MODELOS PREDICTIVOS NO LINEARES

Los modelos no lineales son aquellos en los cuales los parámetros aparecen en forma no lineal. Existen muchos ejemplos de comportamientos no lineales, en todo caso se presenta la esperanza de la variable respuesta Y como una función no lineal de una o más variables independientes.

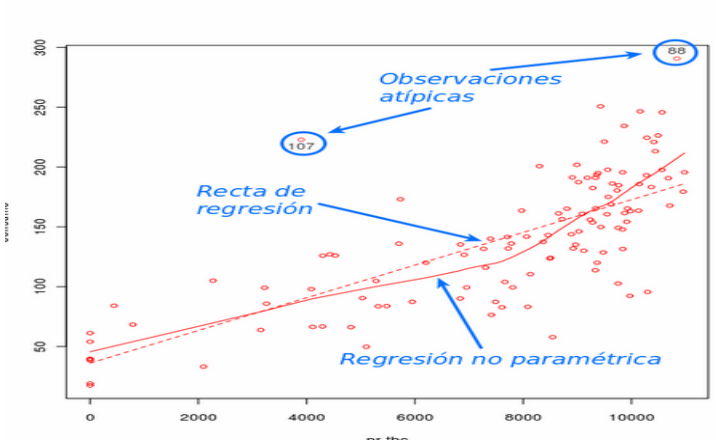


Figura 10. Modelo de regresión lineal frente a una no lineal

[17]http://www.ccee.edu.uy/jacad/2012/x%20area%20y%20mesa/METODOS%20CUANTITATIVOS/Mesa%203/3-Analisis%20de%20datos%20en%20grandes%20dimensiones.%20Estimacion%20y%20seleccion%20de%20variables%20en%20regresion..pdf

2.3.1 Bosques Aleatorios

Es un método de clasificación basado en árboles de clasificación y regresión (CART)ideado por L.Breiman en el 99'. Los bosques aleatorios se han diseñado como mejora del procedimiento CART, dado que es muy inestable y sensible a los datos de entrada, provocando un resultado poco fiable de predicción. Para esto añaden metodología de esamble el cual usa múltiples modelos para obtener la mejor respuesta predictiva que se pueda conseguir a partir de éstos. Concretamente el random forest es un ensamble de árboles de regresión (en nuestro caso) sin poda. Se suelen utilizar cuando se tiene una base de datos de entrenamiento muy grande y con un número alto de variables.

Cada árbol se construye con un subconjunto de datos de entrenamiento aleatorio. A esto se le llama bagging o muestreo por reemplazamiento.

El bagging crea subconjuntos de muestras cogiéndolas del conjunto de entrenamiento de tamaño constante. Algunas observaciones pueden ser repetidas en cada subconjunto. Este tipo de muestras se les llama muestras "bootstrap". Divide las muestras de datos de tal manera que los nodos hijo sean menos heterogéneos que los padre.

Una vez terminada la construcción del árbol, cada hoja corresponde a un subconjunto de las variables X. Así pues, la predicción de este árbol de decisión se representa así:

$$\hat{T}(x,\theta) = \sum_{i=1}^{N} w_i(x,\theta) y_i$$

$w_i(x,\theta)$ Son los pesos que se definen de la siguiente manera: son igual a una constante positiva si Xi es clasificado en la misma hoja, si no entonces es cero. Las constantes se eligen teniendo en cuenta que la suma de los pesos debe ser uno.

Algoritmo

Cada árbol se construye siguiendo el siguiente algoritmo:

1.)Para cada árbol el conjunto de entrenamiento será un bootstrap.

2.)Para crear cada nodo del árbol se usa una parte de las variables del problema de manera aleatoria. La variable más relevante de este subconjunto se usa en el nodo. Durante la expansión del bosque se emplea el mismo número de variables.

3.)Cada árbol se desarrolla hasta su mayor extensión. No se poda (prunning).

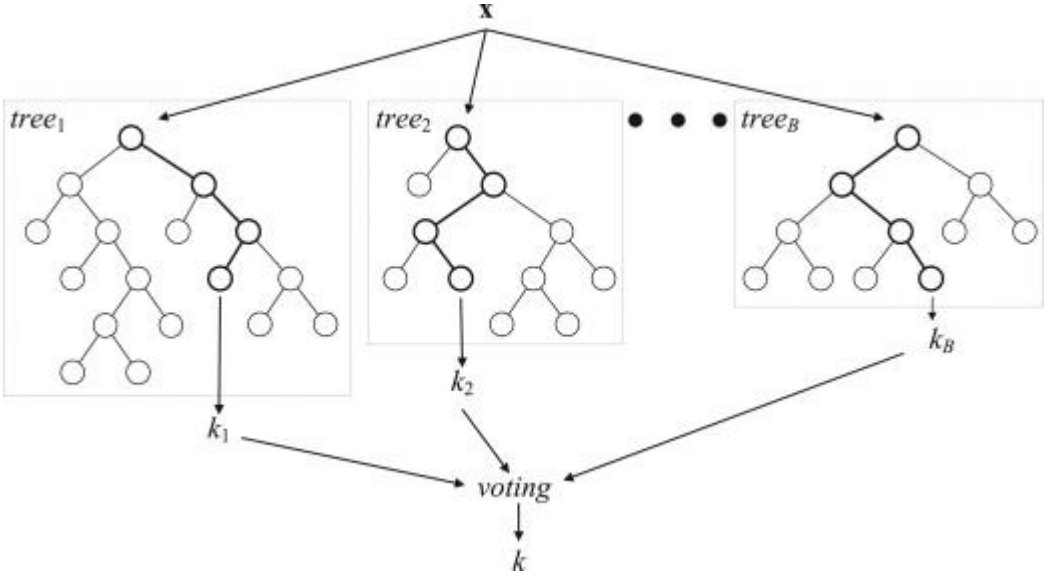
4.) Para predecir una nueva muestra se desplaza el árbol. Se asigna la categoría de muestra de entrenamiento en el último nodo. Este procedimiento es iterativo, y se repite en todos los árboles del conjunto.

El resultado del conjunto de modelos de árboles representa el modelo ensamblado final, donde cada árbol vota por un resultado y el que tenga la mayoría gana. En el caso de un modelo de regresión el resultado es el valor medio del resultado expulsado por cada árbol de regresión.

Según Breiman, el error de predicción del bosque se debe a dos factores:

- La "fuerza" de cada árbol. Un árbol fuerte es aquel que tiene un error bajo tanto de clasificación o de predicción. Cuantos más árboles fuertes se tenga en el bosque, menor será el error de éste.

- La "correlación" entre dos árboles del bosque. Cuánta más correlación haya entre árboles, mayor incremento del error del bosque. Es por esto que Breiman introduce el hecho de elegir un subconjunto de variables elegidas aleatoriamente y de un boostrap de datos, pues reducr la correlación y por tanto el error del árbol.



Ventajas:

- Es el mejor algoritmo de aprendizaje disponible hasta ahora.
- Es rápido para un conjunto de datos elevado.

- Estima qué variables son importantes en la regresión.
- Mantiene sus prestaciones incluso si faltan grandes porciones de información.
- Proporciona información relacionada entre las variables y la regresión.
- Es robusto ante outliers y no necesita que los datos sean previamente normalizados.
- La característica que hace este método muy interesante es la posibilidad de incluir un gran número de variables predictivas en nuestro modelo, generalmente no se necesita hacer una selección de variables antes de empezar a ajustar el modelo.

2.4 BONDAD DE AJUSTE DEL MODELO DE REGESIÓN

La bondad de ajuste de un modelo de regresión se refiere a la calidad del modelo ajustado, si cumple con la representación de las variables implicadas.

Para esto, se emplean diversos métodos.

2.4.1 Suma del error cuadrático

El más común es el conocido como la suma de cuadrados de los errores de predicción (residuales), el cual también se utiliza como criterio para la estimación de los parámetros de regresión de mínimos cuadrados.

$$SCE\ (o\ SC_{Y.X}) = \sum_{i=1}^{n} E_i^2 = \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2$$

Cuanto más pequeño sea su valor, mejor será la predicción que hace.

El único inconveniente de este coeficiente es que el resultado puede variar desde cero hasta cualquier valor positivo. Es decir, no tiene un límite superior, con lo que este índice puede resultar de difícil interpretación en la práctica.

2.4.2 Validación cruzada

La validación cruzada es una herramienta que se emplea en minería de datos, muy útil a la hora de ajustar modelos. Se usa tanto para comprobar la validez del modelo como para comprar muchos de ellos.

Este procedimiento consta de dos fases: entrenamiento y generación de resultados. En la fase de entrenamiento se especifica el número de particiones, por lo tanto se determina cuántos modelos temporales se van a crear. Para cada partición se dividen los datos en un conjunto de prueba y conjunto de test y se entrena cada modelo temporal con el conjunto de prueba. Se obtiene un estadístico por cada modelo.

3

Diseño e Implementación

El objetivo final de este proyecto es proponer un sistema que prediga cuantitativamente el éxito de una canción y además sea capaz de determinar en qué puesto de la lista estará según el puntaje estimado que obtenga.

Se ha decidido enfocar la propuesta como un problema de regresión.

Las muestras ayudarán a modelar una serie de patrones, que van desde modelos lineales hasta no lineales. Estudiando los residuosque cada uno deja, se optará por aquél que produzca menor error. Esto significa que es el que mejor se aproxima a los valores reales.

En este capítulo detallará la metodología seguida, desde la extracción de muestras del servidor Echonest, la partición de éstas para crear el conjunto de entrenamiento y el de test, hastael ajuste de los modelos.

3.1 OBTENCIÓN DE MUESTRAS

Como se explicó en el capítulo de estado del arte, se decidió escoger las mejores 100 canciones de Pop y 76 de Rock según el chart de Youtube, con el objetivo de comprobar si las características más importantes que se determinan como diagnóstico de éxito en la música Pop son las mismas que para el Rock.

Hay que tener en cuenta que la mayoría de canciones de Pop no lo son como tal, se puede encontrar géneros variados, entre ellos el Hip Hop, electrónica… Se mezcla canciones tan variadas como: “Gangnam Style” (electrónica) y “I Knew You Were Trouble” (Country).

Los datos de las muestras que se emplearon (como artista y nombre)están en el capítulo de anexos.

Una de las posibilidades para subir las muestras al servidor de Echonest, y que devolviera los datos que se necesitan era tener los archivos de las canciones guardadas en el ordenador que se estuviera usando. Pero esto sería una tarea engorrosa e innecesaria. El servidor trabaja con muchas compañías que le proveen la mayoría de canciones existentes hoy en día: Spotify, Itunes, EMI y 7digital entre otros. Por lo que se reduce el trabajo a unas pocas canciones que no poseen.

- Canciones **no existentes** en Echonest

Si el servidor no posee alguna pista, se procede a subirla mediante el método upload() con una solicitud http a través de la herramienta cURL.

Como se dijo con anterioridad, la respuesta a la subida de una canción es un proceso asíncrono, por lo que hay que esperar para que Echonest revise si existe esta canción en sus bases de datos, si no está genera un código ID_song.

- Canciones **existentes** en Echonest

Se comprueba canción por canción que el sevidor la tenga con el método search().Seenseñará un ejemplo de la búsqueda de la canción "Diamonds" de la artista Rihanna:

http://developer.echonest.com/api/v4/song/search?api_key=FILDTEOIK2HBORODV&format=json&results=1&artist=rihanna&title=diamonds

Como respuesta devolverá, en formato JSON, cierta información de interés necesaria para posteriores pasos a realizar. Capturaremos el "id" de la canción, es el número de identificación de ésta en el directorio de Echonest. Gracias a él se puede acceder a los metadatos de la canción cuantas veces se quiera.

Este mismo proceso se realiza para cada una de las muestras que se desea tratar.

Una vez obtenido el conjunto, se procede a crear un catálogo, también de formato JSON. El catálogo se utiliza con el propósito de subirlo a la plataforma Echonest y que de forma automática devuelva los datos, que serán los predictores de las muestras en el modelo de regresión.

3.1.1 Creación del catálogo

Echonest permite la gestión decatálogo.Se le da este nombre a una colección de ítems de música (ya sea de artistas o canciones), que pueden ser usados como entrada de parámetro para llamadas a otros métodos del API.

Dichos catálogos se usan para diversas finalidades, como por ejemplo:

- Listas para reproducción.
- Aplicaciones de recomendación de música.
- Solución de flujo: para aplicaciones de usuario que necesitan conocer parámetros como el tempo de todas las canciones del catálogo.

En un principio se debe crear primero el catálogo en la cuenta de Echonest que tenga el usuario, es decir, crear el espacio donde se guardarán las muestras.

Se pasa por parámetro:

- api_key que es la contraseña de usuario para acceder al Api de Echonest.
- -El formato de la respuesta a la solicitud. Siempre será JSON.
- El tipo de catálogo que se quiere crear. Hay de dos tipos: artista o canción.
- Nombre de la etiqueta del catálogo.
- Y la url al que se quiere hacer la petición: siempre será: *"http://developer.echonest.com/api/v4/catalog/create"*.

La respuesta que devuelve inmediatamente consta de:

-El estado de la instancia. Si está correcto y no ha ocurrido ningún fallo en la petición el mensaje de estatus es “Success”.

- El ID del catálogo. Esto es muy importante dado que con este número se gestiona cualquier instrucción que se desee con respecto dicho catálogo.

3.1.2 Actualización del catálogo

Una vez creado el registro se procede a actualizarlo, ya sea añadiendo o eliminando ítems. Cada vez que se actualiza, el contenido del catálogo es resuelto por el Echo Nest IDs.

El cuerpo de la petición tiene la siguiente estructura:

"*ítem id*": nombre del ítem, puede ser cualquiera, pero con caracteres restringidos. En este caso fue la concatenación de nombre del artista y nombre de la canción.

"*song_id*": es el ID de cada canción, el cual se obtiene como se explicó en apartados anteriores.

Al igual que en la creación, la actualización debe enviar una solicitud HTTP por línea de comandos.

El proceso de resolución ocurre asíncronamente a la llamada, por lo que no es instantáneo. Por esto el método update devuelve un "ticket" que puede ser usado para la llamada de estatus para chequear la situación de la actualización.

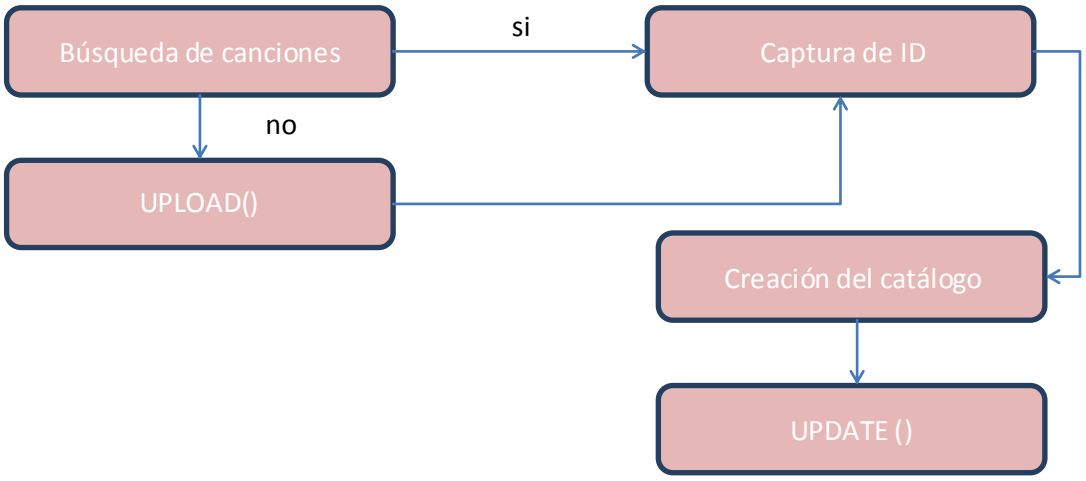


Figura 11. Proceso de obtención de muestras de Echonest

Finalmente estamos a un paso de poder transformar los datos a una matriz, con la que se pueda trabajar en Matlab.

Por lo tanto, llamamos a una simple función *Top_100_features.m*[18] para transportar los datos desde el servidor Echonest a Matlab. A continuación se hace una breve descripción de los métodos que se utilizan:

Parse_json(): convierte objetos JSON en estructuras. Echonest devuelve la información en formato JSON.

18 Para mayor información sobre la función Top_100_features.m dirigirse a los anexos.

Urlread() baja el contenido Web HTML desde una url pasada por parámetro en un string. Sólo devuelve texto.

3.2 PROTOTIPO

El prototipo se compone de 5 fases claramente diferenciadas. Son las siguientes:

1. Inserción de datos en el programa informático para cálculo matemático MATLAB.Creación del conjunto de observaciones (X).
2. Instauración de las matrices de entrenamiento y test [Xtrain, Xtest, Ytrain, Ytest] a partir del conjunto de observaciones.
3. Implementación de los distintos métodos de regresión utilizando como parámetros de entrada la matriz de entrenamiento.
4. Predicción del número de visitas en Youtube (Ytest) de nuevas canciones.
5. Comprobación de la calidad de los modelos.

3.2.1 Inserción de la base de datos

Se explicó en el apartado anterior la extracción de la información del servidor Echonest para luego ser incrustada en el programa Matlab. La matriz de datos queda guardaba como variable *feature_cell*, de dimensiones 100Danceability.

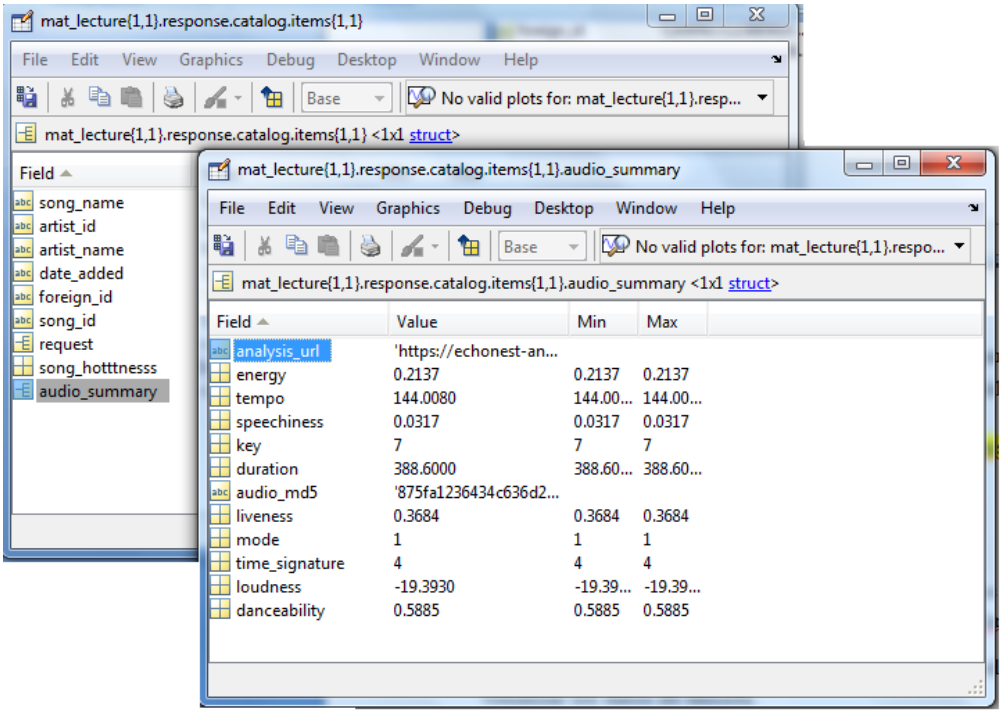
Con el vector de variable independiente Y que llamaremos *score*, no pasa lo mismo, dado que estos datos no se obtienen por medio de Echonest.

Carga de la matriz de características

Se extrae la información del servidor Echonest llamando al método urlread( ). Lo que se le pasa por parámetro es la url con el contenido http de la respuesta al análisis de audio. Al definirse el formato de la respuesta en JSON (no hay otra manera), se debe convertir el formato JSON a .mat para que sea legible en Matlab.

Existe un pequeño código [19]creado por Joel Feenstra que convierte objetos JSON en estructura de datos [parseJson()]. Con esto ya se puede trabajar, dado que se convierte a objetos celda y es como tratar con una matriz numérica.

La extracción de los datos de la estructura `mat_lecture`para convertirla en celda, sin necesidad de utilizar el método *struct2cell( )*, se realiza del siguiente modo:



De manera casi automática, la organización de los datos se plantea como un problema de recorrido de matrices, simplemente usando un bucle for.

Así que simplemente se carga la matriz `feature_cell` mediante el comando *load.m* para visualizar los datos de entrada.

Carga del vector resultado

La manera de cargar el vector score es completamente diferente. No se sacan los datos de Echonest sino de Youtube. Así pues se genera un archivo *Youtube.xls*en el que se insertan el número de reproducciones que tienen la lista de canciones de forma ordenada.

[19]Código parseJson() se encuentra en http://www.mathworks.com/matlabcentral/fileexchange/20565-json-parser/content/parse_json.m

Así que es mucho más fácil, simplemente se importa dicho archivo a Matlab con la siguiente llamada:

score = xlsread('youtube','Hoja1','A1:A99');

3.2.2 Selección del conjunto de entrenamiento y test

La partición de las muestras para la asignación de dichas matrices es una decisión de gran importancia. Hay que tener siempre en cuenta que cuentos más datos de entrada se tengan para modelar,el modelo ajustado será mejor al predecir los datos con los que se entrenó, pero no necesariamente predecirá mejor muestras nuevas que caigan fuera de él (sobreajuste). También es importante elegir muestras para el entrenamiento de todas las calidades, es decir: canciones tanto en los primeros puestos como en los últimos.

Teniendo en cuenta lo dicho anteriormente se procede a distribuir datos.

- 90% serán de entrenamiento= Xtrain, Ytrain.
- 10% serán de test = Xtest, Ytest.

De forma automática, se genera esta distribución con la llamada de *gendat()*, método obtenido de PROTOOLS[20].

>>muestras=[feature_cell score];% metemos las dos matrices en una.

>> [tra, tst] = gendat(muestras, 0.8); % generación de datos aleatorios con herramienta PRTOOLS, partiendo datos de entrenamiento (90%) y datos de test(20%)

>> Ytrain=double(tra(:,10));

>> Xtrain=double(tra(:,1:9));

>> Xtest=double(tst(:,1:9));

>> Ytest=double(tst(:,10));

[20] PRTOOLS: Toolbox de Matlab sobre reconocimiento de patrones para representación y generalización. http://prtools.org/

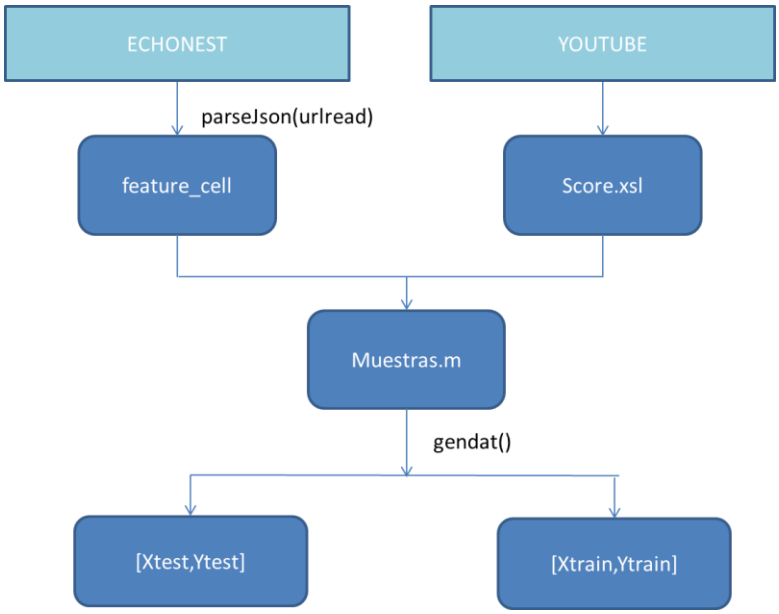


Figura 12. Diagrama de flujo obtención de muestras

3.2.3 Modelos

La implementación de diversos modelos de regresión será uno de los estudios más interesantes de este proyecto. Con una elección correcta del método de regresión óptimo para la obtención de una predicción más fiable, podremos saber si una canción será un éxito y en qué rango aproximado de las listas va a estar.

Dicho esto, nos centraremos en modelos lineales y no lineales. El rango de búsqueda va a ser lo más extenso posible, dado que no hay una relación/distribución clara entre los observables y los predictores si los visualizamos en una gráfica.

La salida de cada regresor devolverá los coeficientes estimados del vector de pesos. Una vez ajustada la función que describe el éxito de una canción se podrá predecir la "victoria" de cualquier canción que se desee.

3.2.3.1 Modelo lineal

Se procede a obtener los coeficientes de regresión empleando mínimos cuadrados bajo regresión lineal.

Para minimizar el error de estimación y garantizar unos resultados que no dependan de la repartición de los datos entre entrenamiento y test se aplica validación cruzada, por esto se obtiene un vector de regresión (bi) por cada iteración. En total son nueve. Estas iteraciones son el producto de dividir en k particiones el conjunto de datos. Una de estas particiones se utiliza como conjunto de validación y el resto como conjunto de entrenamiento.

El objetivo de la validación cruzada es el elegir el modelo cuyos coeficientes minimicen el error de predicción. Por esta razón lo más lógico es obtener el MSE promedio de cada modelo ajustado con respecto a la iteración usada como test en cada caso y utilizar aquel vector de pesos que generemenor error.

MSE_i= media[($ypred_i$-$ytest_i$).^2]

	MSE
Iteración 1	3,9E+16
Iteración 2	3,79E+15
Iteración 3	3,203E+14
Iteración 4	1,44E+15
Iteración 5	9,5E+14
Iteración 6	1,01E+15
Iteración 7	2,56E+15
Iteración 8	1,85E+15
Iteración 9	3,41E+15

Tabla1 . MSE por iteración

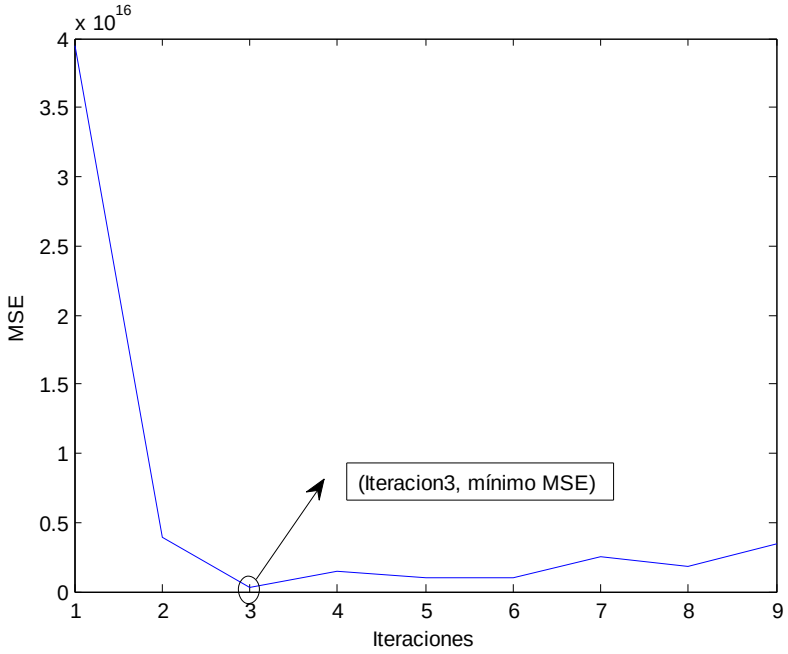


Figura 13. Relación entre error cuadrático medio por iteración

De esta manera, se capturan los coeficientes de regresión de la iteración 3, la de menor error cuadrático medio, y se procede a generar el modelo lineal cuya forma es:

y ~ 1 + x1 + Tempo + Speechiness + Key + Duration + Liveness + Mode + Loudness + Danceability

Hay que recalcar que se modela la función f(x) con todos los predictores de la iteración 3.En la tabla que viene a continuación se detallan los valores estimados de los coeficientes y algunos valores estadísticos de interés como el valor p (*p-value*)[21]. El coeficiente *Intercept* es el valor independiente de la función.

Número de observaciones	90
RMSE	7.05e+07
R.^2	0.0778
R.^2 Ajustado	-0.026
Valor p	0.663

Tabla 2. Estadísticos del modelo lineal

[21] "La probabilidad de obtener un resultado al menos tan extremo como el que realmente se ha obtenido"

Predictor	*Coeficiente estimado*
Valor independiente	-2,14E+06
Energy	-49243143,52
Tempo	276127,8521
Speechiness	-113340846
Key	1635440,055
Duration	-164999,3836
Liveness	37726910,33
Mode	-957987,4386
Loudness	177138,6425
Danceability	111550195,5

Tabla 3. Coeficientes $\hat{\beta}$

3.2.3.2 Modelo Lineal + Stepwise

Se procede a obtener los coeficientes de regresión empleando regresión lineal con ajuste por stepwise.

Como se explicó en el capítulo anterior, stepwise elimina los predictores menos relevantes (más redundantes) y que no aportan ninguna información nueva al modelo.

Para minimizar el error de estimación se aplica validación cruzada. Se utilizan las mismas iteraciones que en el caso anterior.

Se obtiene el MSE de cada modelo ajustado con respecto a la iteración usada como test en cada caso y se captura aquel que tenga menor error.

MSE_i= media[($ypred_i$-$ytest_i$).^2]

	MSE
Iteración 1	3,9E+16
Iteración 2	4,15E+15
Iteración 3	6,73E+13
Iteración 4	1,83E+14
Iteración 5	1,17E+15
Iteración 6	1,04E+15
Iteración 7	2,14E+15

	MSE
Iteración 8	1,68E+15
Iteración 9	2,36E+15

Tabla 4. MSE por iteración

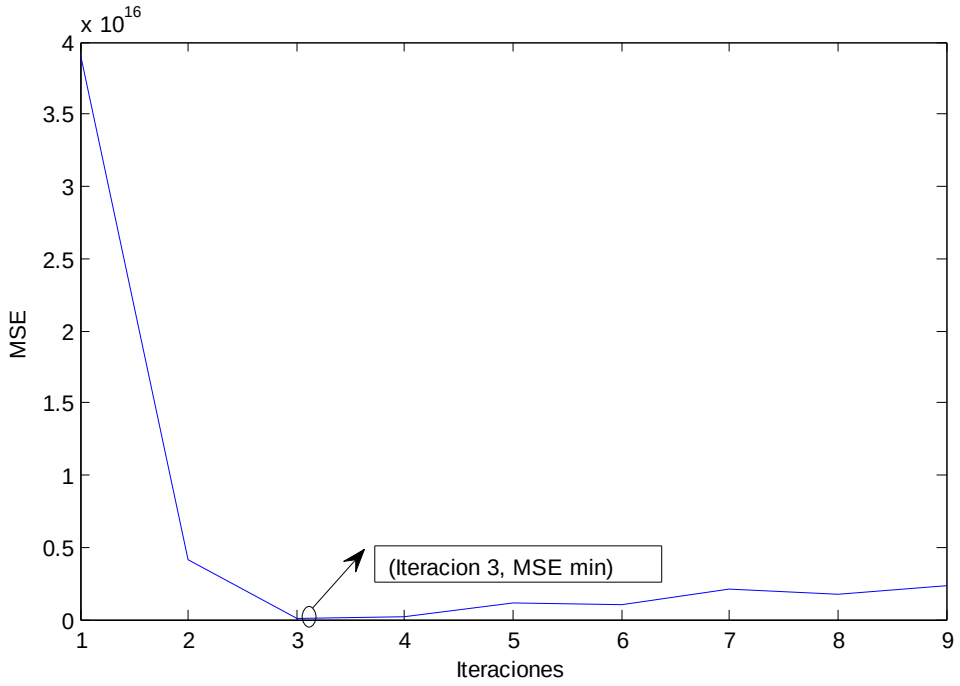


Figura 14. Relación entre error cuadrático medio por iteración

De esta manera, se captura los coeficientes de regresión de la iteración 3, la de menor error cuadrático medio, y se procede a generar el modelo lineal cuya forma es:

y ~ 1

Elimina los predictors: Energy, Tempo, Speechiness, Key, Duration, Liveness, Mode, Loudness y Danceability. Vuelve nulas todas las características definidas, esto significa otras características que no se han contemplado en este estudio son más relevantes que estas a la hora de obtenery.

Si lo analizamos, conceptualmente no tiene ningún sentido. Claramente este tipo de método no sirve para estimar el número de visitas en youtube.

Número de observaciones	90
RMSE	6.96e+07

Tabla 5. Estadísticos del modelo lineal + Stepwise

Predictor	*Coeficiente estimado*
Valor independiente	3,78E+11

Tabla 6. Coeficientes $\hat{\beta}$

3.2.3.3 Modelo Lineal + Robust

Se procede a obtener los coeficientes de regresión empleando regresión lineal con ajustepor robust.

Para minimizar el error de estimación se aplica validación cruzada. Se utilizan las mismas iteraciones que en el caso anterior.

Se obtiene el MSE de cada modelo ajustado con respecto a la iteración usada como test en cada caso y se captura aquel que tenga menor error.

MSE_i= media[($ypred_i$-$ytest_i$).^2]

	MSE
Iteración 1	4,09E+16
Iteración 2	5,61E+15
Iteración 3	1,3E+15
Iteración 4	1,67E+14
Iteración 5	8,37E+13
Iteración 6	1,11E+14
Iteración 7	2,69E+14
Iteración 8	3,75E+14
Iteración 9	4,27E+14

Tabla 7. MSE por iteración

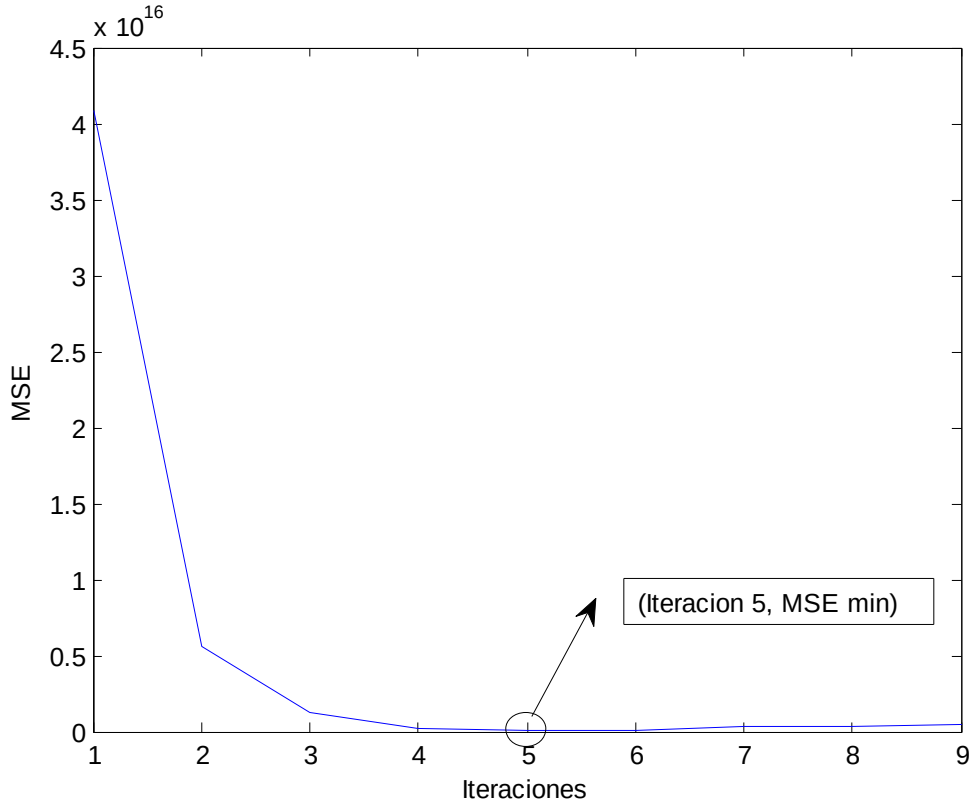


Figura 15. Relación entre error cuadrático medio por iteración

De esta manera, se captura los β de la iteración 5, la de menor error cuadrático medio, y se procede a generar el modelo lineal cuya forma es:

y ~ 1 + x1 + Tempo + Speechiness + Key + Duration + Liveness + Mode + Loudness + Danceability

Número de observaciones	90
RMSE	5.32e+07
R-2	0.175
R-2 Ajustado	0.0822
Valor p	0.066

Tabla 8. Estadísticos del modelo lineal + ajuste robusto

Predictor	*Coeficiente estimado*
Valor independiente	2,21E+07
Energy	-31959166,37
Tempo	79250,14951
Speechiness	13870272,01
Key	612300,6958
Duration	-21299,74988
Liveness	42051173,44
Mode	-1997585,216
Loudness	-358209,0415
Danceability	-12213460,16

Tabla 9. Coeficientes $\hat{\beta}$

3.2.3.4 Regularización Lasso

Como se explicó en apartados anteriores, Lasso en una técnica de regularización. Se aplica el método *lasso*para:

- Reducir el número de predictores.
- Identificar predictores de gran relevancia.
- Genera una estimación de la salida con errores de predicción potencialmente menores que con el método de mínimos cuadrados.

Devuelve coeficientes ajustados por mínimos cuadrados para un conjunto (λ) de coeficientes de regularización .Por defecto ofrece una variedad de 100 coeficientes λ distintos. Por lo tanto la matriz β es de dimensión 9x100.

Al igual que con los anteriores modelos, aplicamos validación cruzada. En este caso, como se obtienen 9x100 coeficientes β por iteración se calcula el error cuadrático medio de cada modelo lasso ajustado utilizando cada uno de los coeficientes. Paso seguido se hace un promediado del MSE de todas las iteraciones por cada parámetro λ utilizado. Se utiliza aquel λ cuyo MSE promedio sea el menor de todos. Sabiendo λ obtenemos el vector de pesos adecuado.

Siendo el vector de pesos:

$$\hat{\beta} = (X^T X + \lambda I)^{-1} X^T y$$

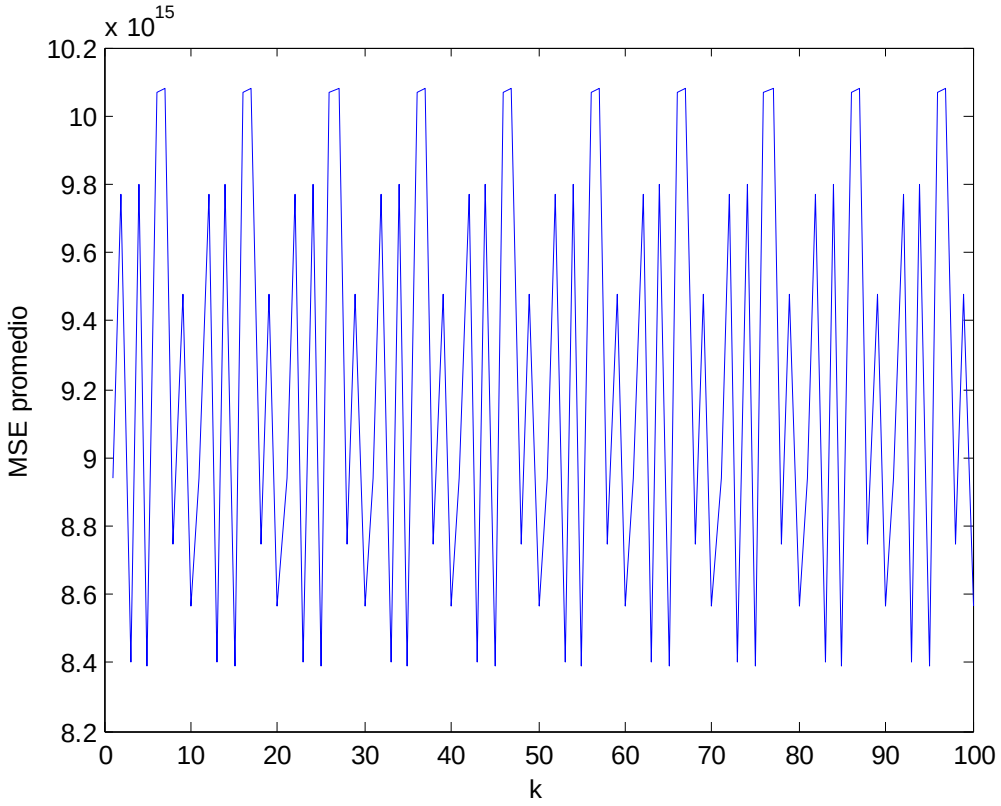


Figura 10. *Relación entre error cuadrático promedio y k.*

Así pues se sabe a qué iteración corresponde, y de aquella se elige el valor de λ cuyo MSE es el menor. Obtenemos el λ y por lo tanto su vector β correspondiente.

	Coeficiente estimado
Valor independiente	0
Energy	0
Tempo	4032,59952965293
Speechiness	-33255107,0181604
Key	223379,985501437
Duration	-18466,8467282325
Liveness	0
Mode	0
Loudness	0
Danceability	72865141,2656357

Taba 11. *Coeficientes $\hat{\beta}$*

Elimina, al igual que Stepwise, algunas variables que no son relevantes para el modelado. Lasso elige las características: Tempo, Speechiness, Key, Duration,y Danceability..

3.2.3.5 Regularización Ridge

Como se explicó con anterioridad, a diferencia de Lasso, el método ridge no anula variables de X como lo hace Lasso. Deja todos los predictores en el modelo.

Así pues, se dispone a obtener los diferentes coeficientes de regresión. El método ridgedevuelve una matriz β gorro con distintas combinaciones, todas dependen del parámetro λ (parámetro ridge). Toma valores entre 0 y 1.

Así que el problema está en averiguar el parámetro k óptimo, interesa que sea aquel que menor MSE devuelva. Para obtenerlo entra en juego la validación cruzada.

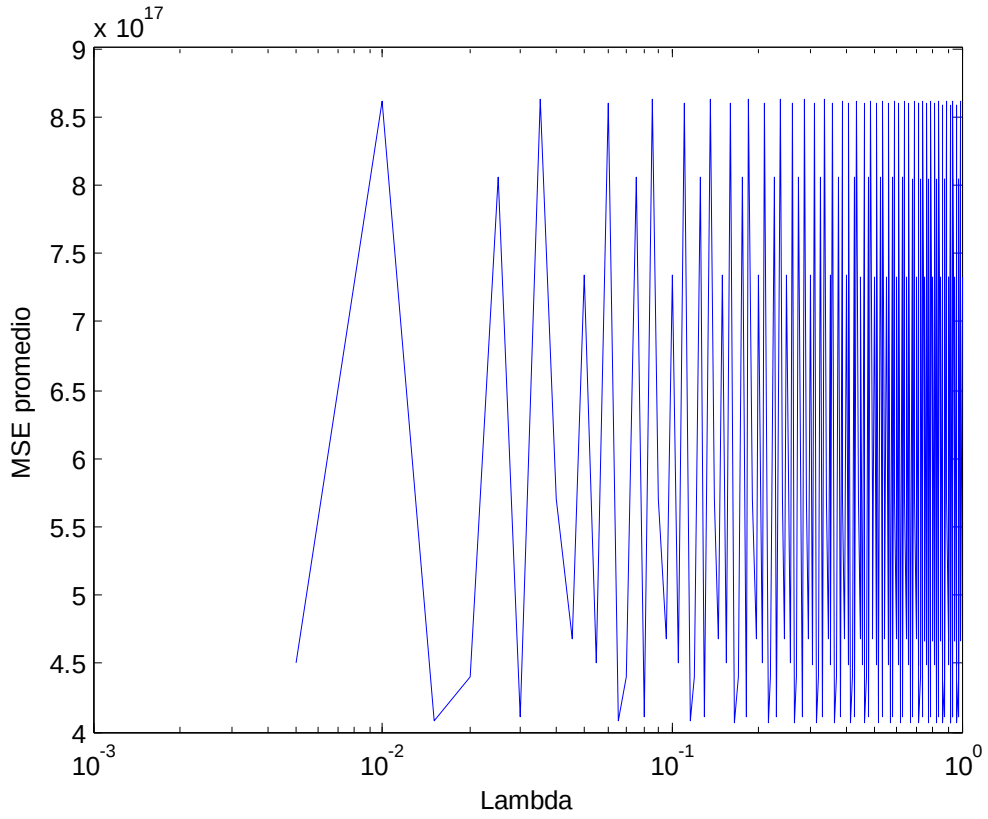


Figura 17. Relación logarítmica entre error cuadrático promedio y lambda

Se selecciona el β cuyo MSE es el mínimo, este corresponde a un valor de k=0,965.

Siendo β ridge:

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$$

Calculamos β reemplazando cada valor por sus correspondientes, dando el siguiente valor del estimador:

Y el modelo quedaría como:

	Coeficiente estimado
Valor independiente	0
Energy	-49721687,2337938
Tempo	269659,228578768
Speechiness	-105794146,358189
Key	1606808,40056870
Duration	-173087,226192076
Liveness	39371064,7891846
Mode	-490566,890422868
Loudness	233792,826555175
Danceability	111846661,554583

Tabla 12. Coeficientes $\hat{\beta}$

3.2.3.6 Bosques Aleatorios

La metodología que se sigue es crear un objeto TreeBagger por cada tamaño de hoja (por cada p). p puede tomar los siguientes valores: [1 5 10 20 50 100]. Por defecto N=50. Se escoge aquel bosque aleatorio que consigue minimizar el MSE de entrenamiento.

Utilizamos el mismo método de validación que en los modelos anteriores, obteniendo los siguientes errores cuadráticos medios:

Tamaño de la hoja	***MSE promedio***
1	4,68e+16
5	4,80e+16
10	4,98e+16
20	4,93e+16
50	4,92e+16
100	4,95e+16

Tabla 13. MSE por iteración

Como se observa en la tabla, aquel bosque aleatorio de cincuenta árboles de una hoja es el modelo que devuelve el menor error cuadrático promedio utilizando validación cruzada.

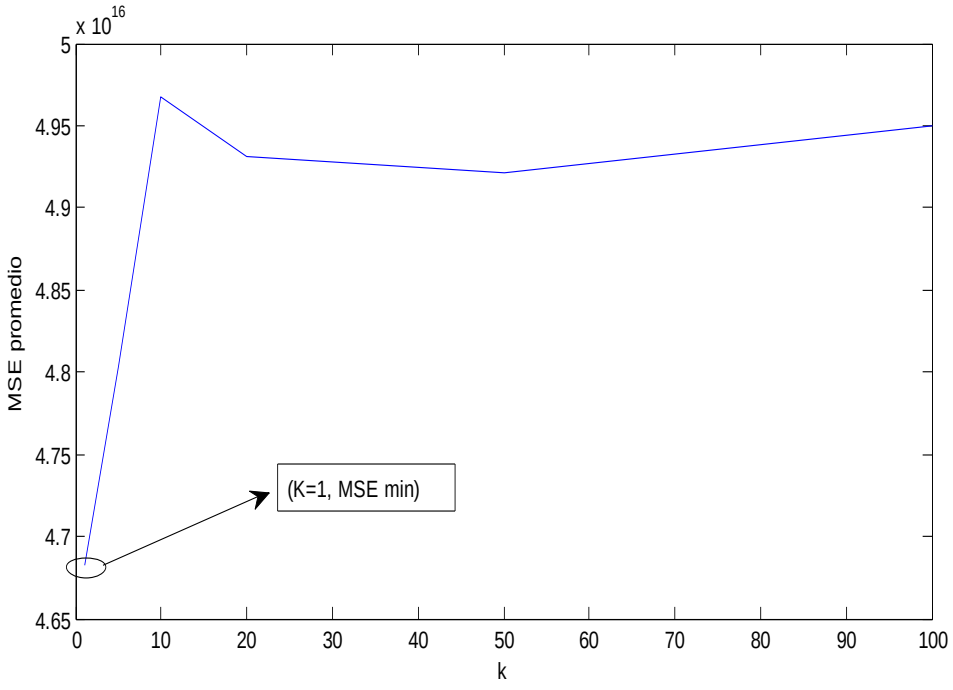


Figura 18. Relación entre error cuadrático promedio y k

En la siguiente figura, se observa el comportamiento del error de entrenamiento del bosque con un número de árboles fijo (50) variando el tamaño de la hoja.

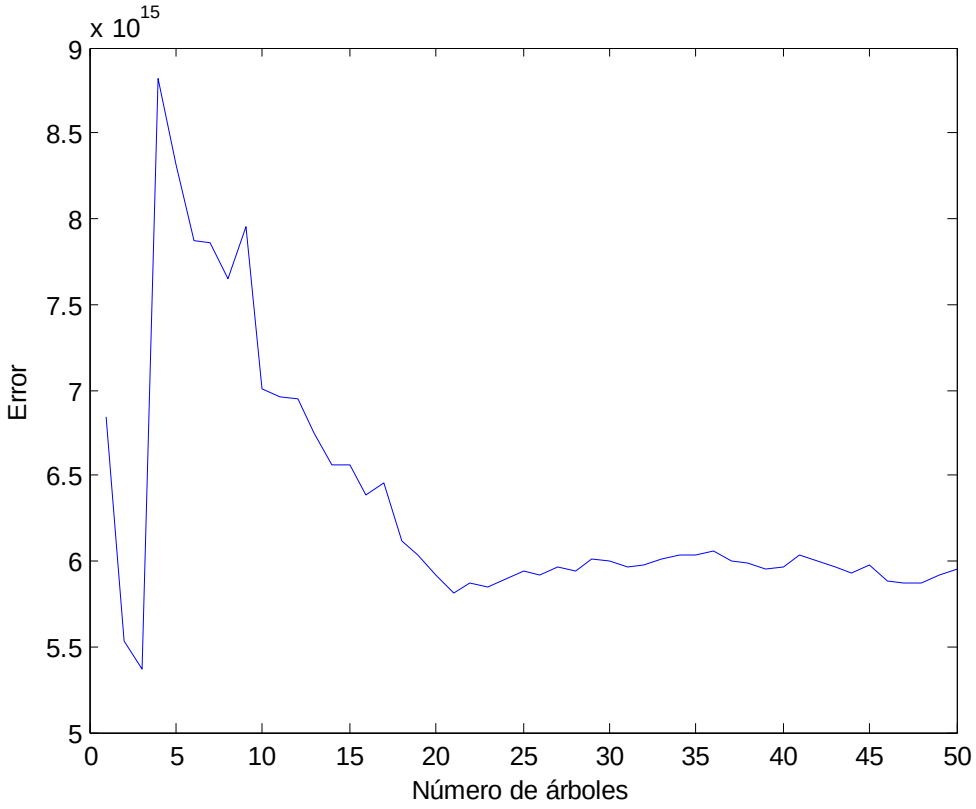


Figura N. *Relación entre error cuadrático medio del conjunto entrenamiento y el número de árbolesen el bosque,*

Diagrama de flujo. Descripción de la metodología

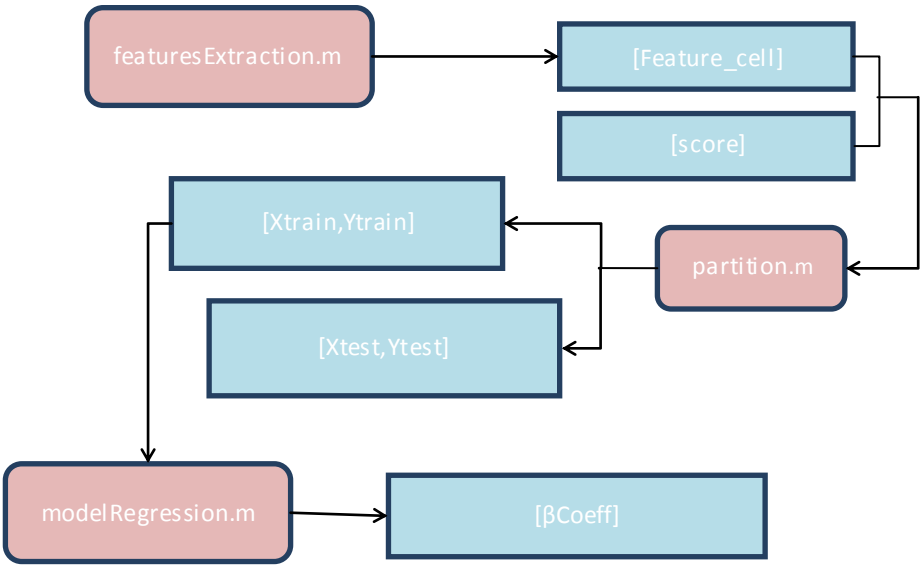


Figura 20. *Esquema parcial del prototipo*

3.2.4 Predicción

En este apartado se verificará cuál es el modelo ajustado que mejor predice el éxito de canciones Pop. El objetivo, como ya se ha expuesto con anterioridad, no sólo se centra en predecir, sino en averiguar si los mismos modelos son adaptables a otros géneros y lo más importante: si los pesos que se le da a ciertas características en la música Pop son los mismos que para otra.

El contraejemplo es con la música Rock. Se utilizarán como matriz de entrada muestras de canciones de esta variedad musical. La tabla de relación de las canciones se encuentra en los anexos.

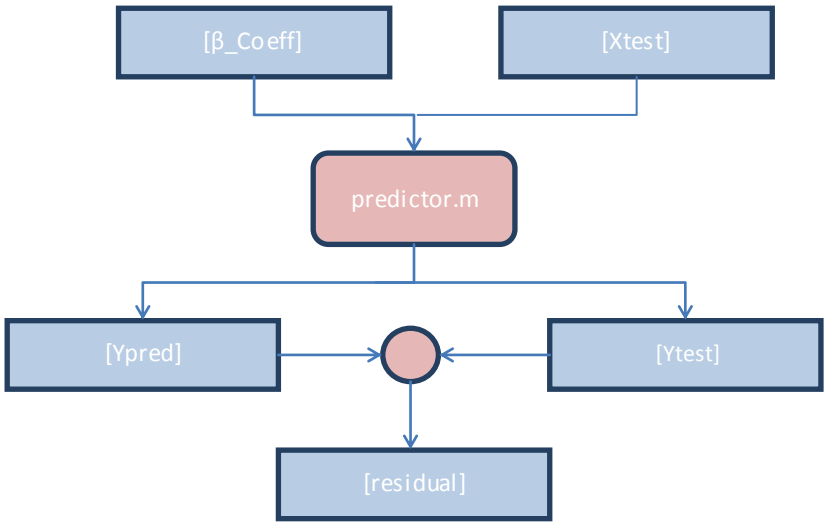


Figura 21. Diagrama de predicción

Aquí se utiliza el mismo método de predicción para:

- Modelos Lineales.
- Modelos Lineales + Stepwise.
- Modelos Lineales + Robust.
- ModeloTreeBagger

Introducimos el modelo devueltos por los métodoy la variable Xtest, que son las características de las canciones restantes de la partición, como entrada del método predict (). Como su nombre indica, predice la respuesta del modelo a una variable nueva de entrada.

```
>>      ypred = predict(modelo,Xtest);
```

Devuelve un vector de dimensión 29x1.

Una vez obtenida la predicción del "éxito", se calculará la suma del error cuadrático, donde el error es la diferencia entre el valor estimado y el real.

$$\hat{\epsilon} = \vec{y} - X\hat{\beta} = (I - X(X'X)^{-1}X')\vec{y} = (I - X(X'X)^{-1}X')\vec{\epsilon}$$

Como se ha dicho con anterioridad, la suma de cuadrados de los residuales puede oscilar entre 0 y cualquier valor positivo. Si este sumatorio da 0, el modelo de regresión se ajusta perfectamente a los datos; cuanto mayor sea su valor, ello significará que más erróneas son las predicciones de la ecuación de regresión y, por lo tanto, peor su bondad como modelo predictivo.

Predicción para modelo de regresión Ridge

Para la predicción de nuevas variables con estos dos modelos de regularización se aplica que la salida estimada es igual al producto entre el vector de coeficientes de regresión por la muestra de entrada:

$$Ypred = b * Xnew$$

El cálculo del error es el mismo que para los anteriores modelos.

A continuación se expondrá una tabla con los errores cuadráticos medios de predicción de cada uno de los modelos anteriormente vistos, para proceder a elegir el de mayor fiabilidad y confianza.

Modelo	***MSE_train***	***MSE_test***
TreeBagger	9.91e+14	1.30e+17
Lasso	4,57e+15	1,37e+17
Stepwise	4,79e+15	1,38e+17
Regress	4.43e+15	1.42e+17
Robust	5.21e+15	1.43e+17
Ridge	4,42e+15	1,41e+17

Tabla 14. Tabla de orden ascendente con los valores de MSE de cada modelo

El valor marcado con color azul es el menor, correspondiente al modelo de bosques aleatorios.

Es importante afirmar que si se comparan los errores tanto de test como de entrenamiento, estos últimos en todos los casos, son mucho menores que los de validación, por lo que ningún modelo cae en sobreajuste.

4

Valoración Experimental

En este capítulo se evaluará la calidad del modelo elegidocon la ayuda de gráficas y tablas. La discusión de los resultados se centrará en valorar para qué género musical se alcanzan predicciones de éxito más fiables con los diversos algoritmos.

Los apartados en los que está dividido son:

- **Relaciones de dependencia** entre :
    1. Las características de información musical para saber cuáles de ellas son dependientes y cuáles no.
    2. El número de visitas en Youtube y las distintas características de las muestras.
- **Evaluación de la calidad de los modelos** teniendo en cuenta los residuosde:
    1. Conjunto de entrenamiento
    2. Conjunto test
- Obtención de las **muestras predichas** con el mejor modelo ajustado.
- **Listas del ranking**. Elaboración de una lista ordenada de canciones respecto al número de visitas predichas en Youtube.

4.1 ANÁLISIS A PRIORI DE LOS DATOS OBTENIDOS

En este apartado se analizarán las relaciones de dependencia lineal entre las distintas variables predictoras y con respecto al vector de salida.

4.1.1 Canciones Pop

En la siguiente gráfica se enseña la relación que tiene la salida con cada una de las variables de entrada.

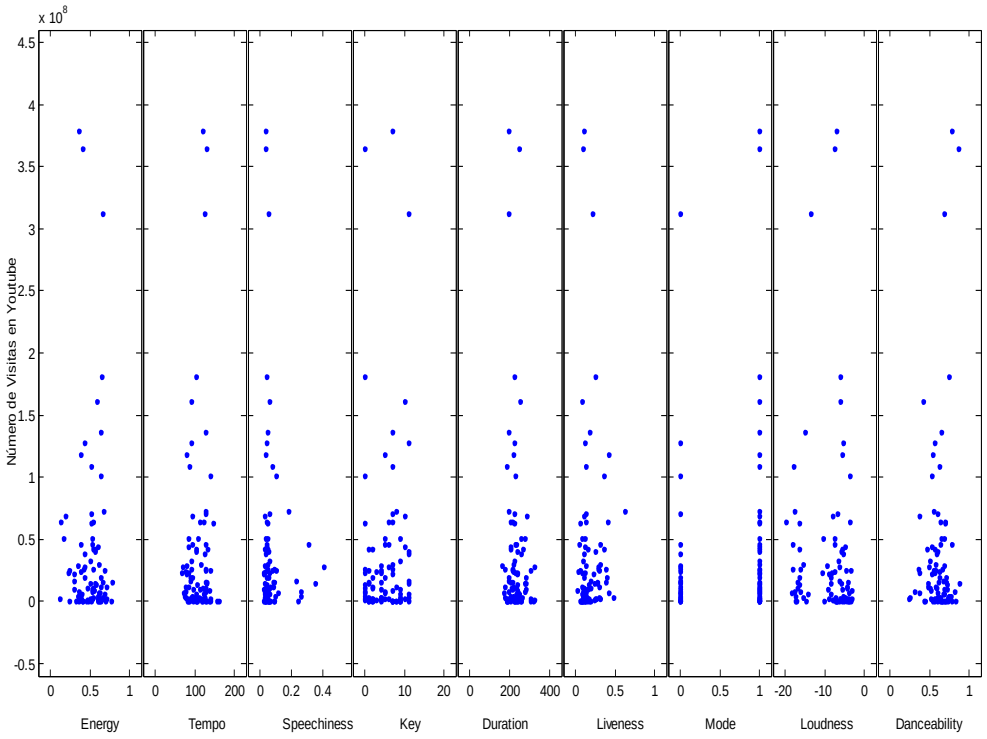


Figura 22. Relación que guarda el éxito en Youtube y cada uno de los predictores

Con esta gráfica se destaca:

1. El tempo varía generalmente entre los 80-130 bpm en la mayoría de canciones.
2. El puntaje está fuertemente vinculado a valores cercanos a cero de speechiness. Esto significa, como bien se ha dicho en la explicación de los atributos acústicos, que

distingue que son pistas instrumentales, sin diálogo hablado. Lo cual tiene mucho sentido.

3. Gran parte de la música de muestra está compuesta en escalas en modo mayor (el 70%). El otro 30% son escalas menores. Musicalmente hablando quiere decir que gustan más generalmente melodías y ritmos alegres. El modo menor confiere una armonía melancólica.
4. No hay una relación clara entre el número de visitas en youtube y la tonalidad en la que están compuestas las piezas, pero se sabe que casi un 30% están en Reb (Do#) y otro 20% en Si.
5. No tienen una duración superior a los 320 s, ni menor de los 160 s.
6. La mayoría de estas pistas no están grabadas en vivo, el valor de liveness tiende a ser bajo. Es lógico, dado que es música con un potente post procesado.

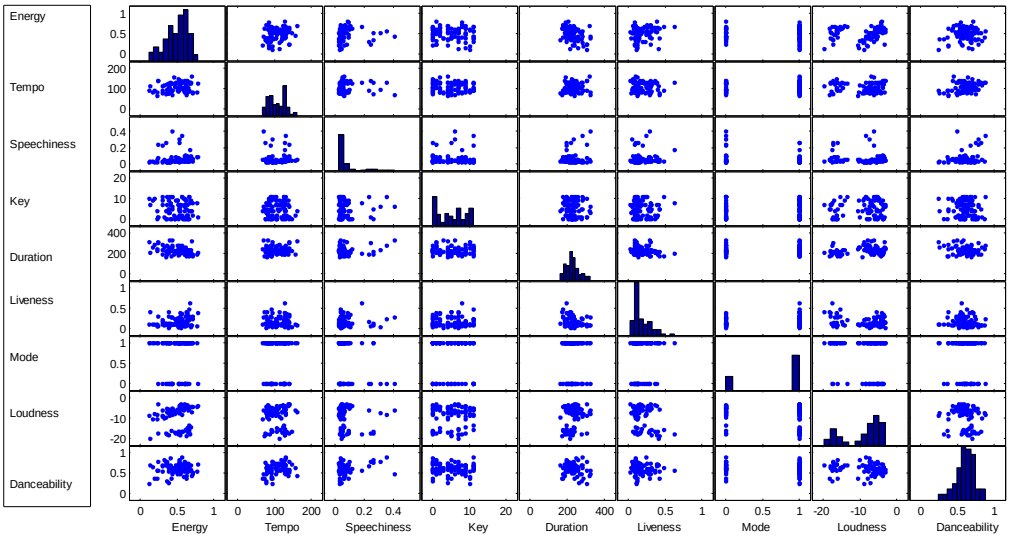


Figura 23. Diagrama de dispersión entre predictores.

De esta figura se puede abstraer lo siguiente:

1. Todas las variables son independientes y totalmente incorreladas de la variable mode (modo) y de la variable speechiness. Hay que tener en cuenta que estas sólo toman dos valores: 0 ó 1.
2. El resto más o menos tiene una dependencia, claramente no es lineal, tampoco se puede sacar a simple vista su relación.

4.1.2 Canciones Rock

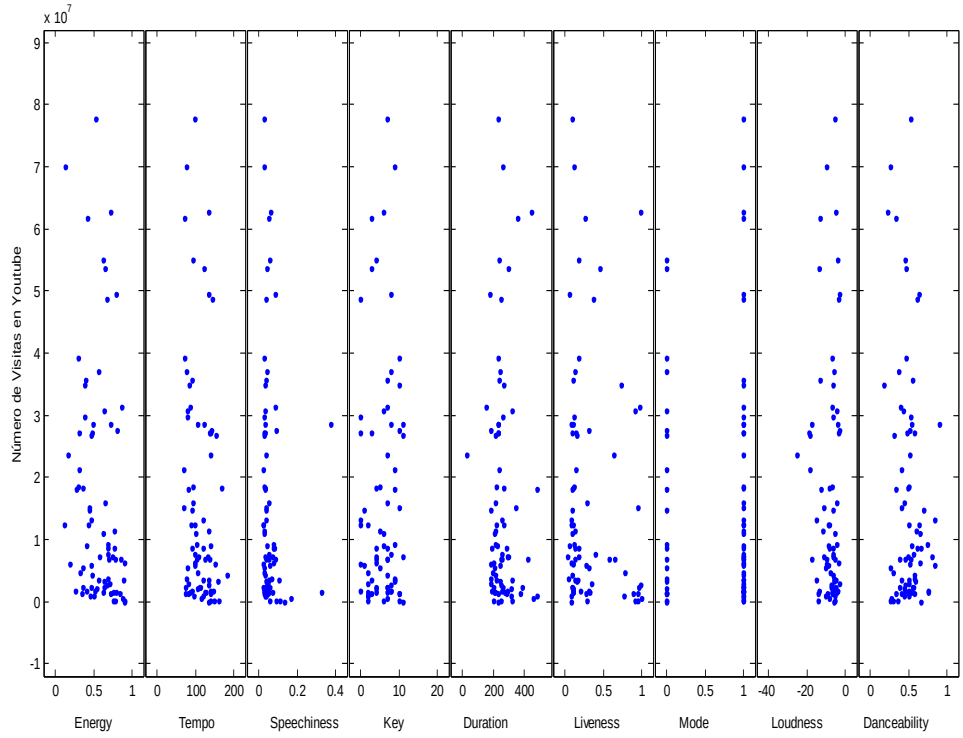


Figura 24. Relación entre la salida y cada uno de los predictores.

Volvemos a repetir la misma gráfica pero esta vez con muestras de música Rock. Se denotan varias características de diferenciación con respecto a las de Pop y otras muy relacionadas.

1. El intervalo de Tempo es más amplio que en el caso de las muestras de Pop. El rango se encuentra entre los 70 y los 180 bpm. Alcanza valores mucho mayores, por lo que el ritmo es más rápido.
2. El valor de speechiness tiende a cero, por lo que reconoce que hay instrumentos y no es pista hablada.
3. No hay una clara distribución de la tonalidad predominante:
   - Un 17% está en Fa # o Solb
   - Un 14.5% en La.
   - Un 11,8% en Do.
   - Un 11,8% en Re#.

4. En cuanto a la duración predominante varía entre los 200 y 250 segundos.
5. La mayoría de muestras (casi un 46%) están grabadas en estudio, pero a diferencia con las de pop otra proporción grande de las canciones están grabadas en vivo.

6. Al igual que con el pop, suelen ser composiciones en modo mayor (65%), el resto menores.
7. La sonoridad es parecida a la de pop, casi un 40% tiene -5 dB.

4.2 PRESENTACION DE RESULTADOS

A continuación se procederá a ilustrar diferentes gráficas para comprender visualmente lo que ocurre con cada modelo a la hora de entrenar y validar cada uno de ellos. Y lo más interesante, ver aquellas muestras que no se ajustan correctamente.

Gracias a éstos y al error cuadrático que arrojan en la predicción será mucho más fácil elegir el método óptimo para la finalidad del proyecto.

Los tipos de gráficas que se mostrarán son las siguientes:

1. Relación entre el número de visitas en Youtube predichas y reales de cada canción, tanto de entrenamiento como de validación.
2. Relación del error cuadrático y las muestras de test.
3. Comparativa de errores cuadráticos por muestra test de todos los modelos empleados en el caso de Pop.

4.2.1 Entrenamiento con muestras Pop

Por cada modelo se representa una gráfica de barras que relaciona el número de visitas predichas y reales en Youtube de cada una de las canciones utilizadas para entrenar dicho modelo.

4.2.1.1 Modelo Lineal

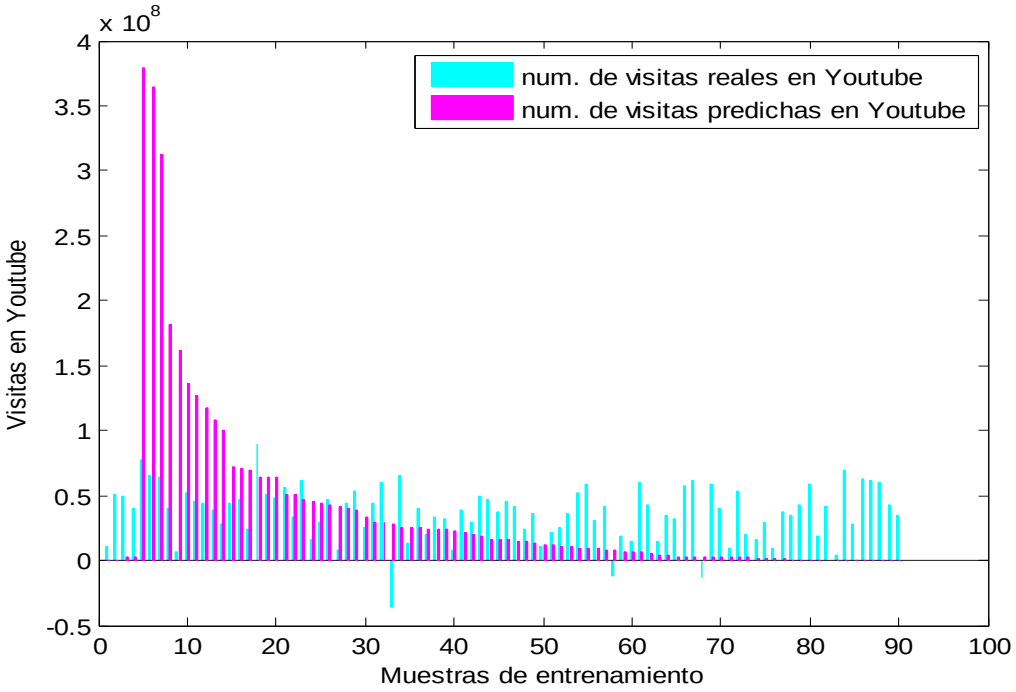


Figura 25. Numero de Visita predichas y originales de muestras de entrenamiento

4.2.1.2 Modelo Lineal. Stepwise

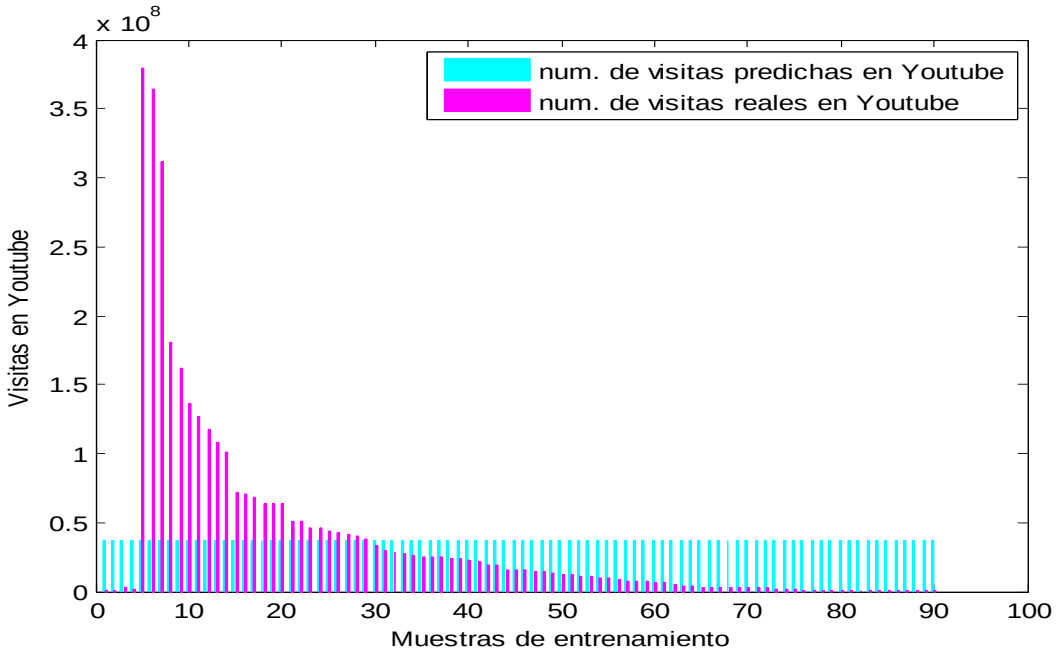


Figura 26. Numero de Visita predichas y originales de muestras de entrenamiento

4.2.1.3 Modelo Lineal. Robust

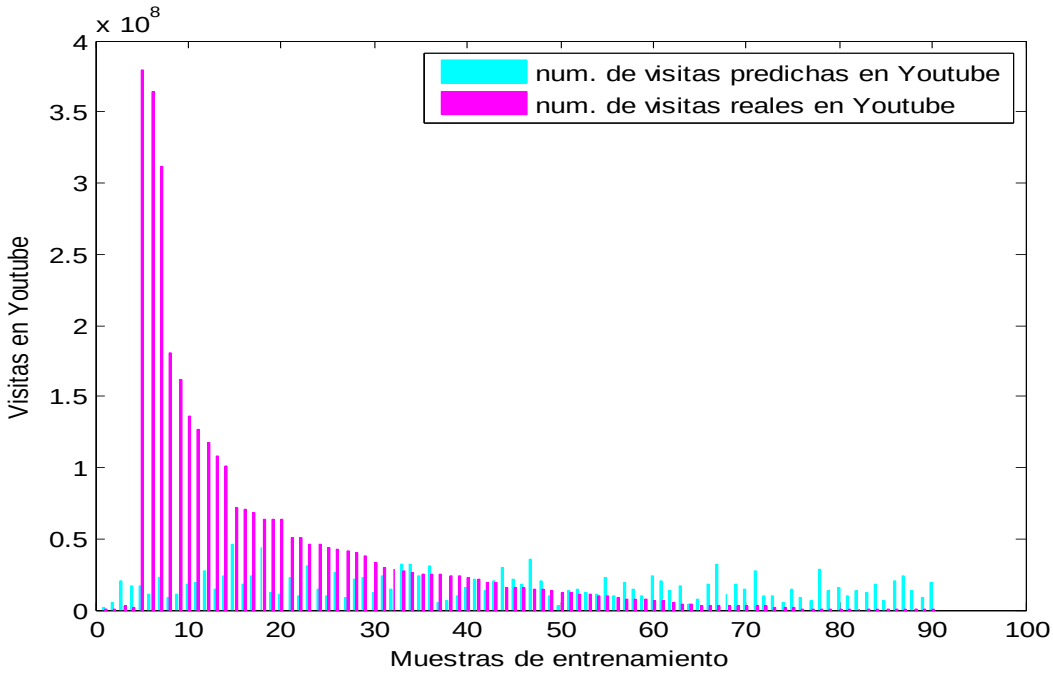


Figura 27. Numero de Visita predichas y originales de muestras de entrenamiento

4.2.1.4 Modelo Lasso

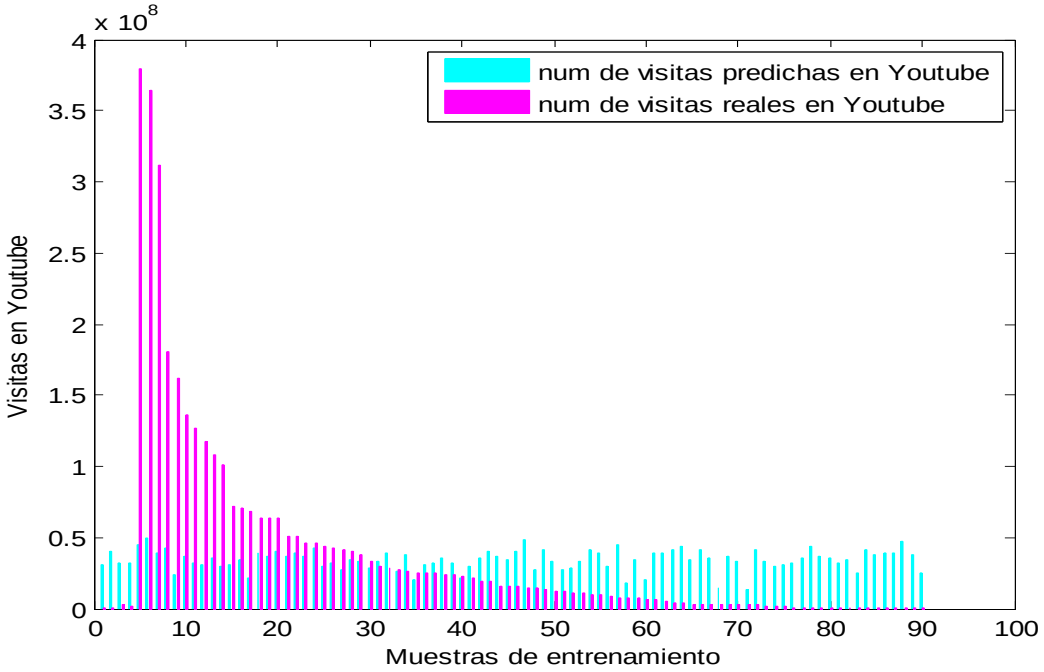


Figura 28. Numero de Visita predichas y originales de muestras de entrenamiento

4.2.1.5 Modelo Ridge

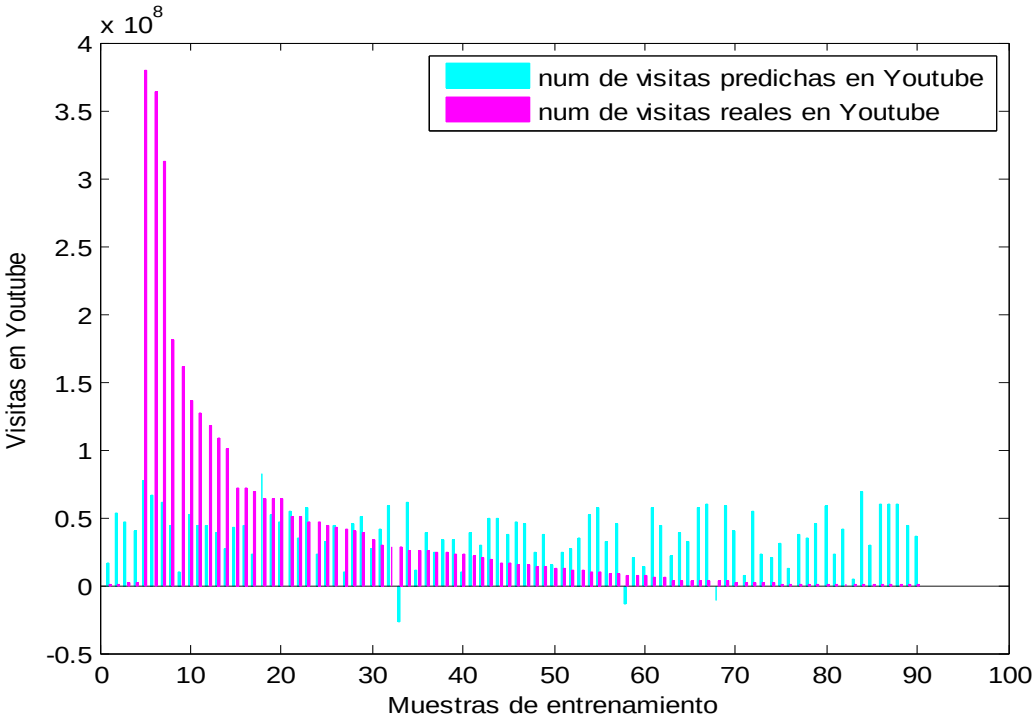


Figura 29. Numero de Visita predichas y originales de muestras de entrenamiento

4.2.1.6 Bosques Aleatorios

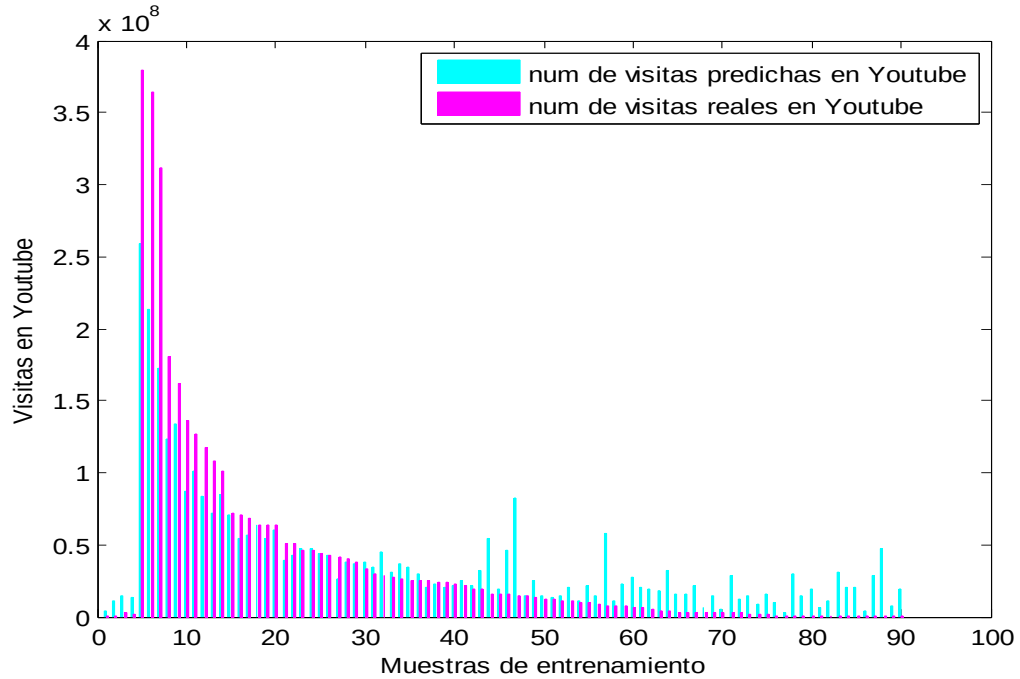


Figura 30. Numero de Visita predichas y originales de muestras de entrenamiento

4.2.2 Validación con muestras Pop

Por cada modelo se representa una gráfica de barras que relaciona el número de visitas predichas y reales en Youtube de cada una de las canciones Pop utilizadas para validar dicho modelo.

Al mismo tiempo se muestra el error de predicción de cada una de las canciones utilizadas.

Por último, se comparan los residuos por muestras generados por todos los modelos de predicción utilizados en una única gráfica de barras.

4.2.2.1 Modelo Lineal

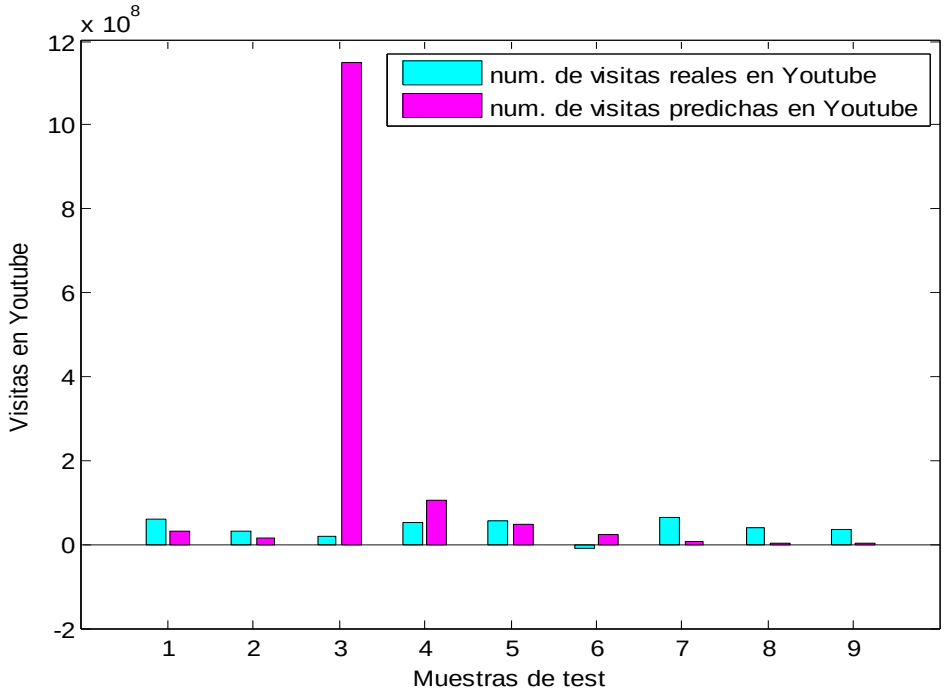


Figura 31. Numero de Visita predichas y originales de muestras de test

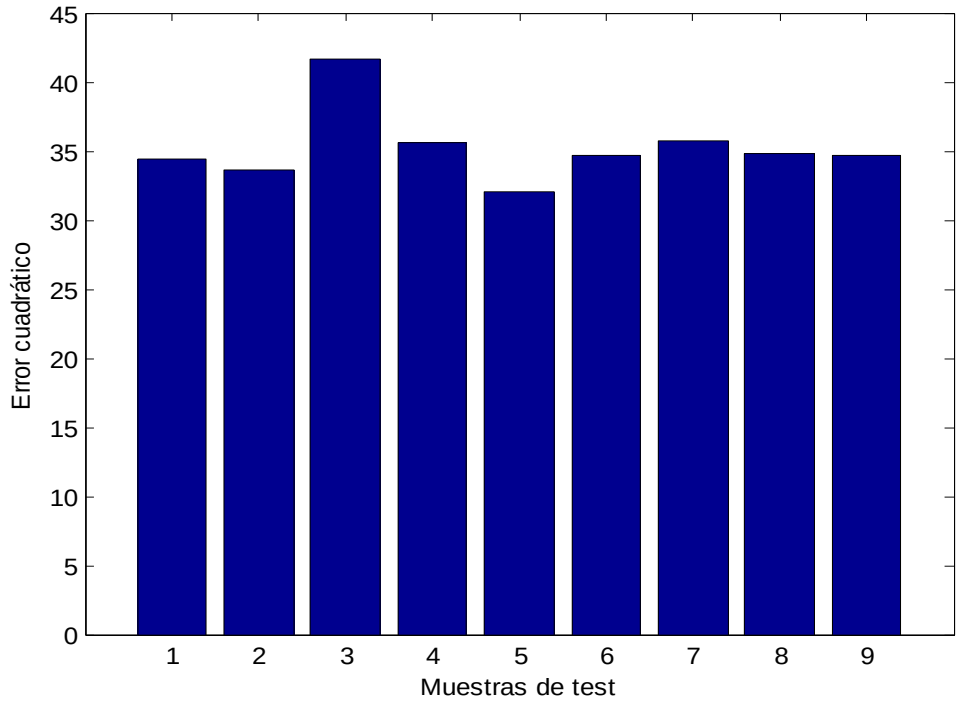


Figura 32. Logaritmo del error cuadrático por muestra

4.2.2.2 Modelo Lineal. Stepwise

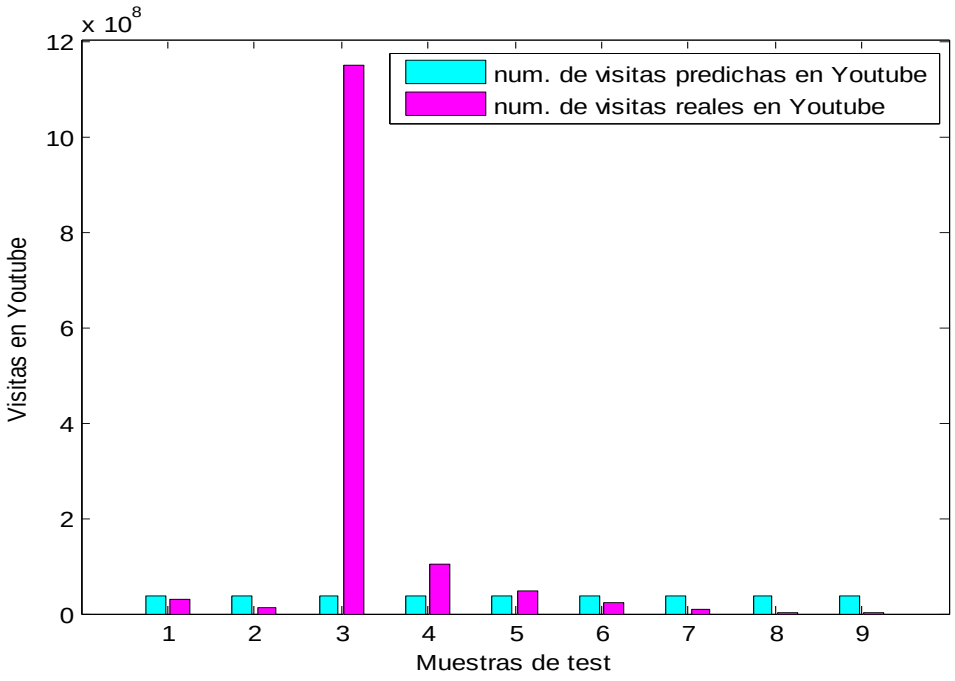


Figura 33. Numero de Visita predichas y originales de muestras de test

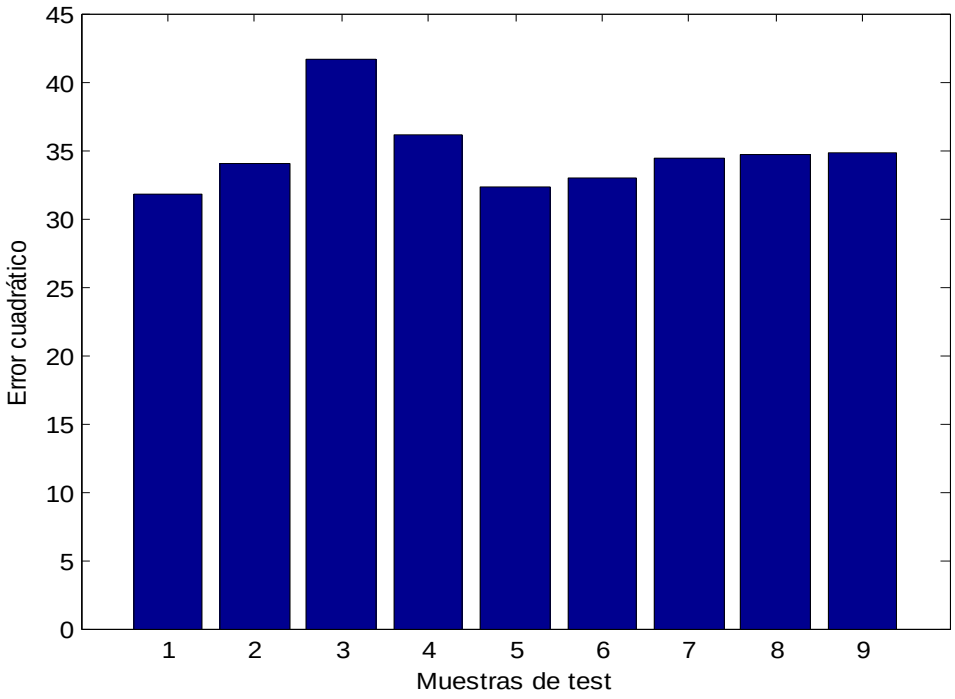


Figura 34. Logaritmo del error cuadrático por muestra

4.2.2.3 Modelo Lineal. Robust

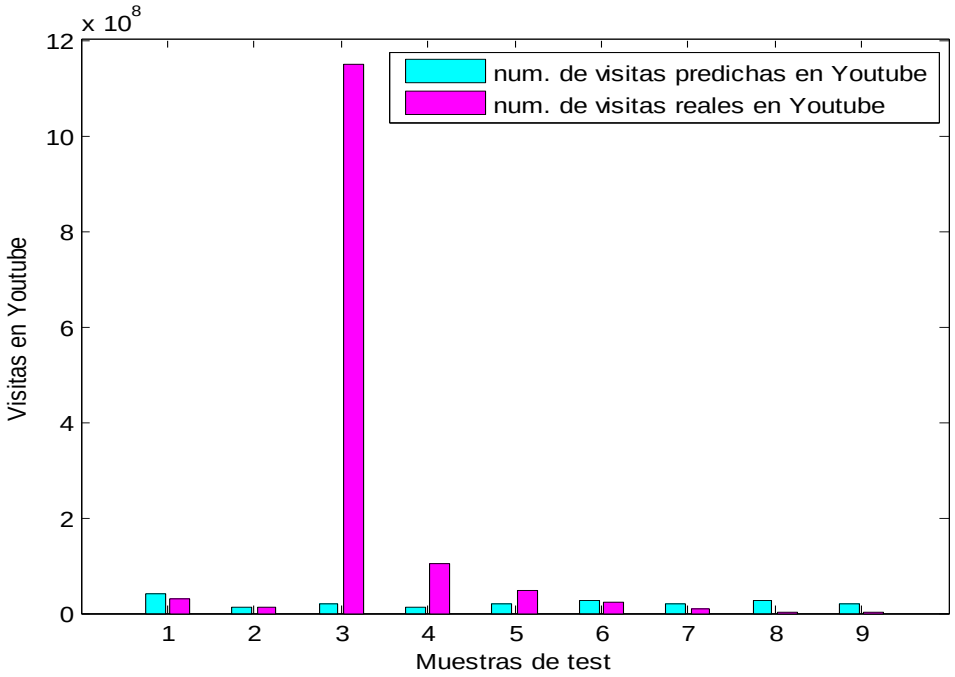


Figura 35. Numero de Visita predichas y originales de muestras de test

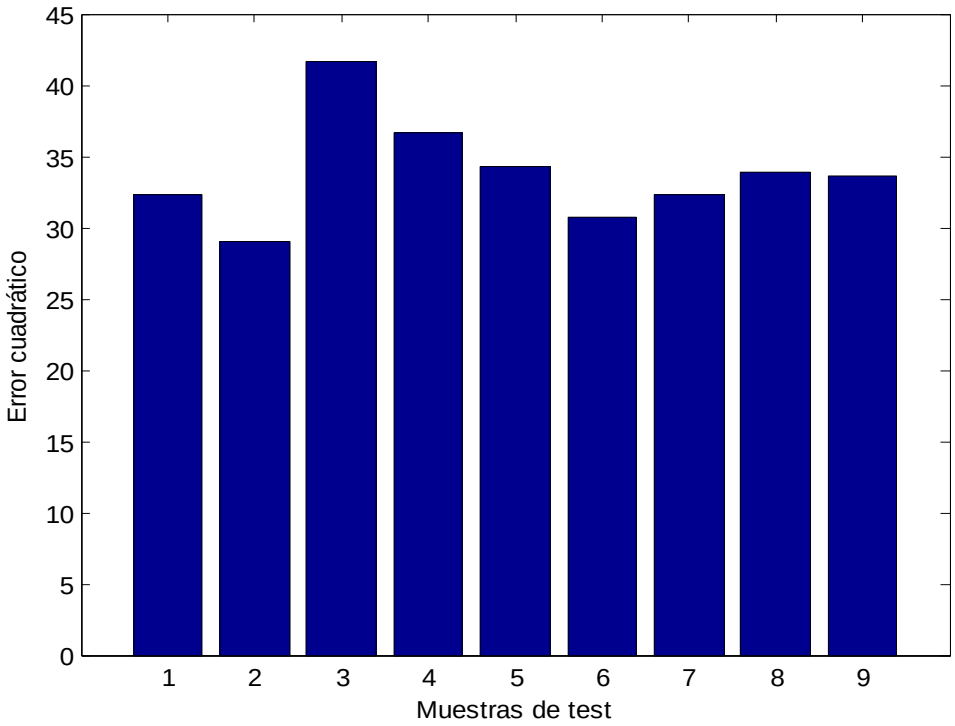


Figura 36. Logaritmo del error cuadrático por muestra

4.2.2.4 Modelo Lasso

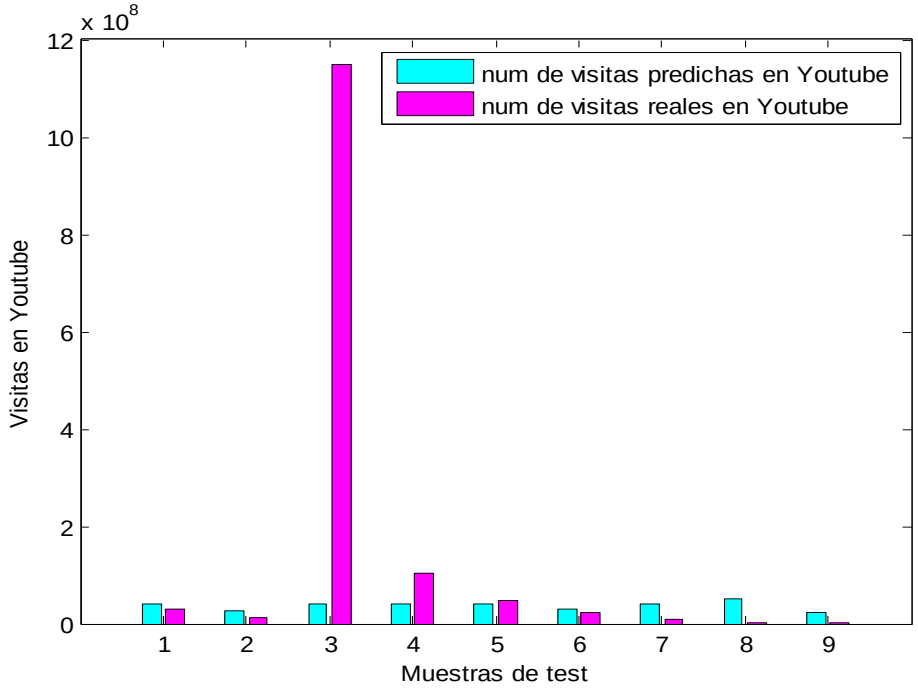


Figura 37. Numero de Visita predichas y originales de muestras de test

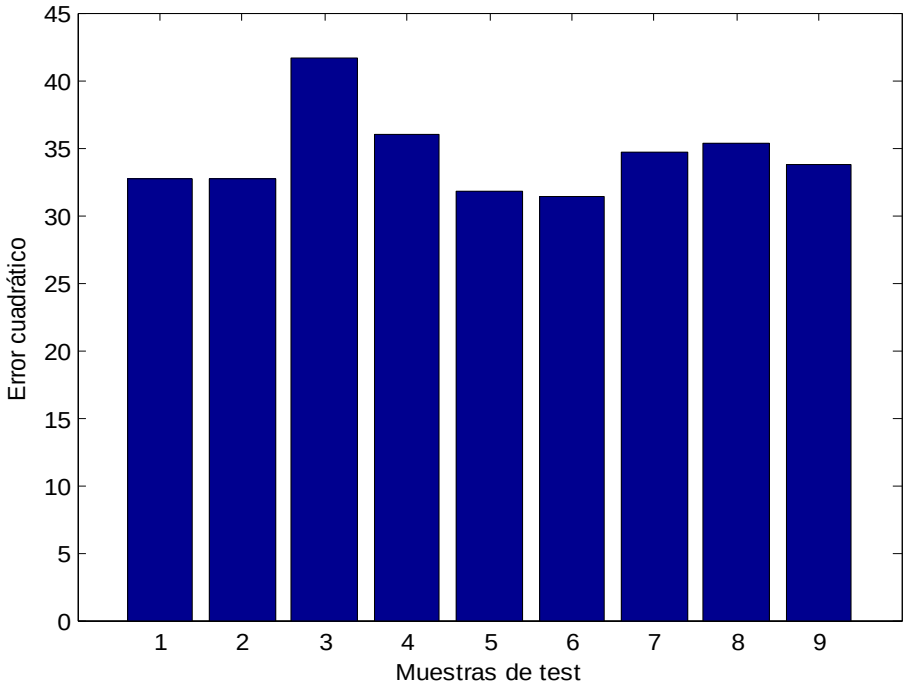


Figura 38. Logaritmo del error cuadrático por muestra

4.2.2.5 Modelo Ridge

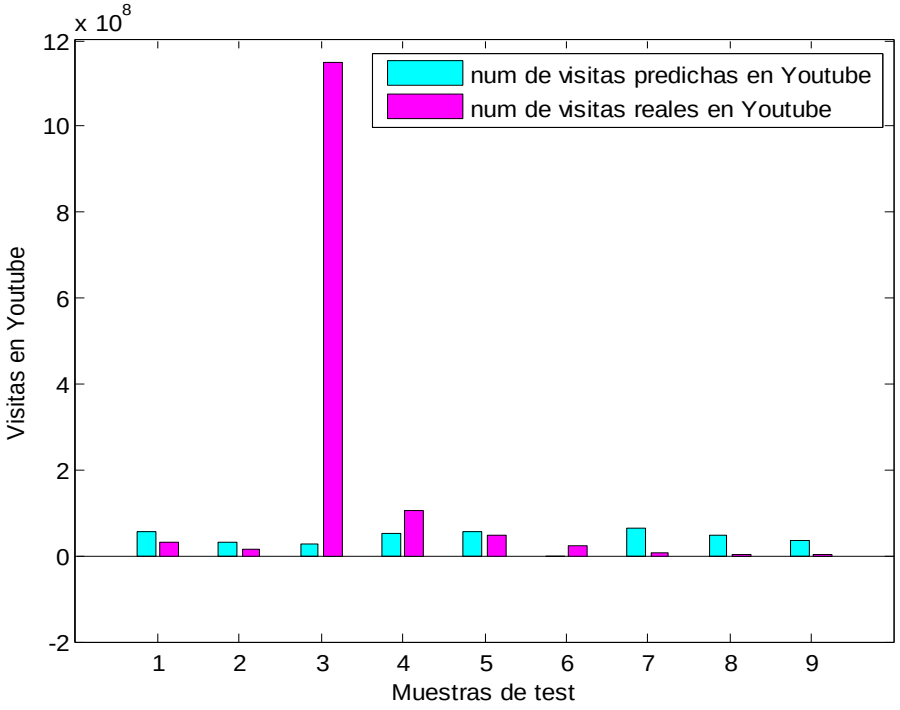


Figura 39. Numero de Visita predichas y originales de muestras de test

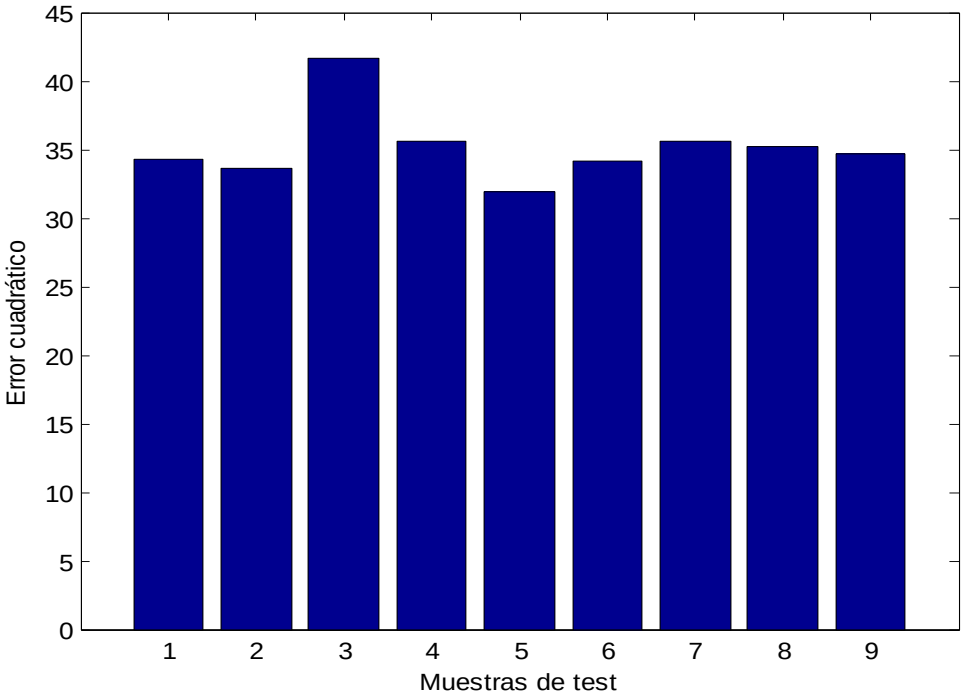


Figura 40. Logaritmo del error cuadrático por muestra

4.2.2.6 Bosques Aleatorios

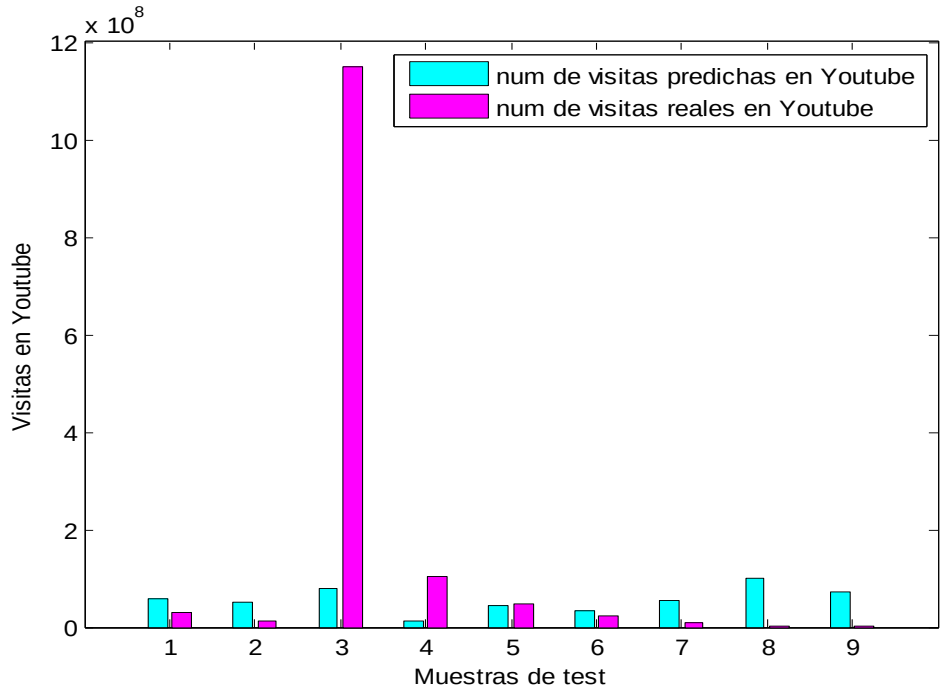


Figura 41. Numero de Visita predichas y originales de muestras de test

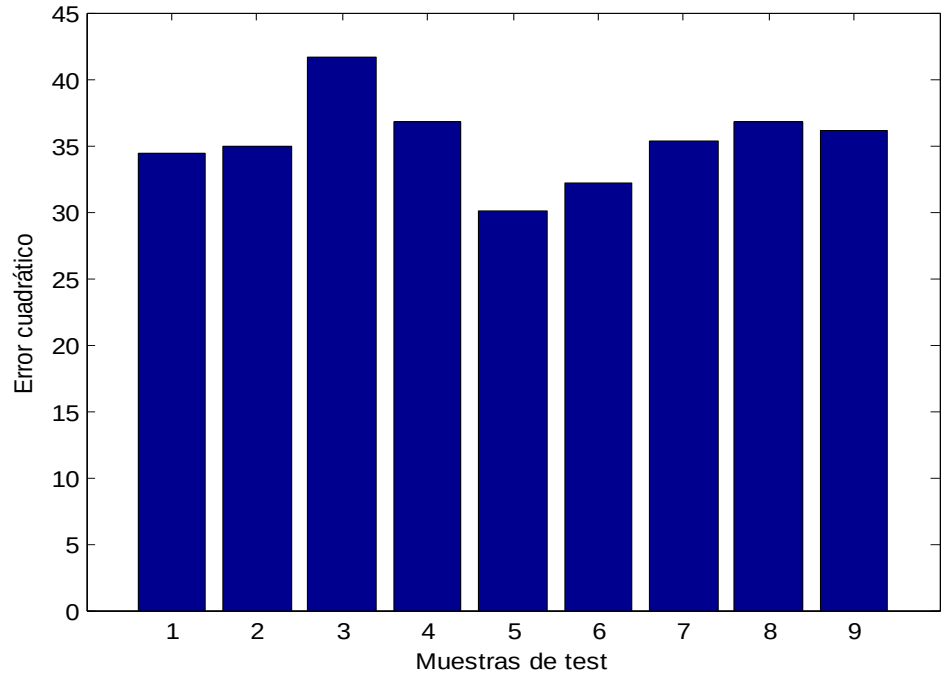


Figura 42. Logaritmo del error cuadrático por muestra

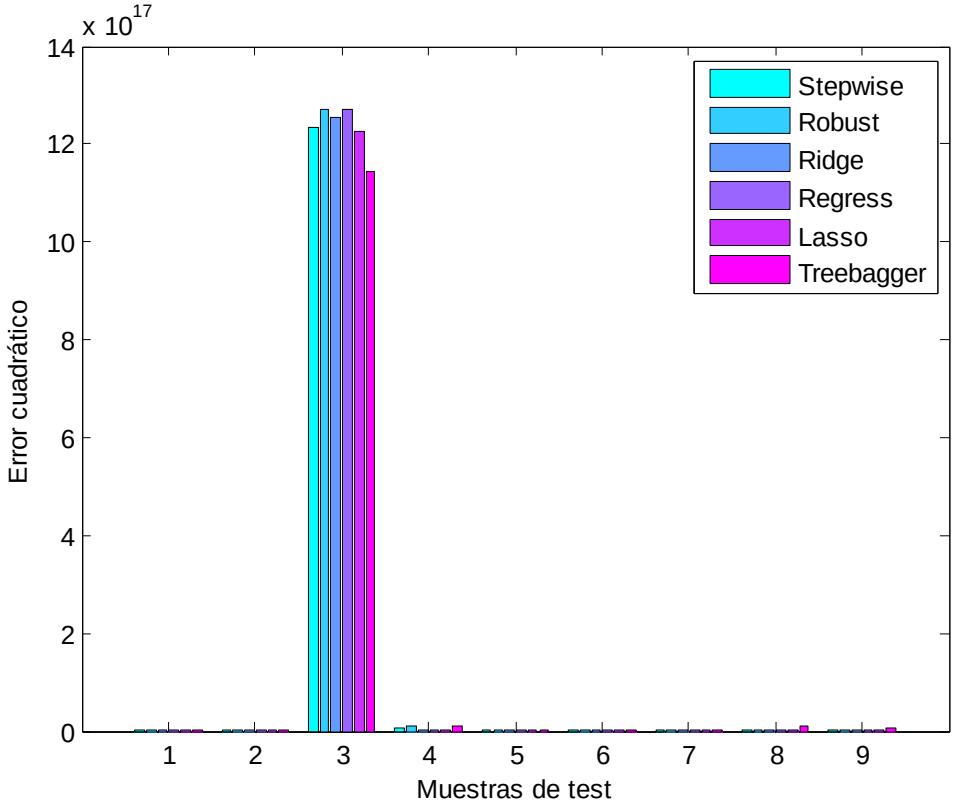


Figura 43. Diagrama de barras agrupadas de los residuos de los diferentes modelados

4.2.3 Validación con muestras Rock

Por cada modelo se representa una gráfica de barras que relaciona el número de visitas predichas y reales en Youtube de cada una de las canciones de género Rock utilizadas para validar dicho modelo.

Al mismo tiempo se muestra el error de predicción de cada una de las canciones utilizadas.

4.2.3.1 Modelo Lineal

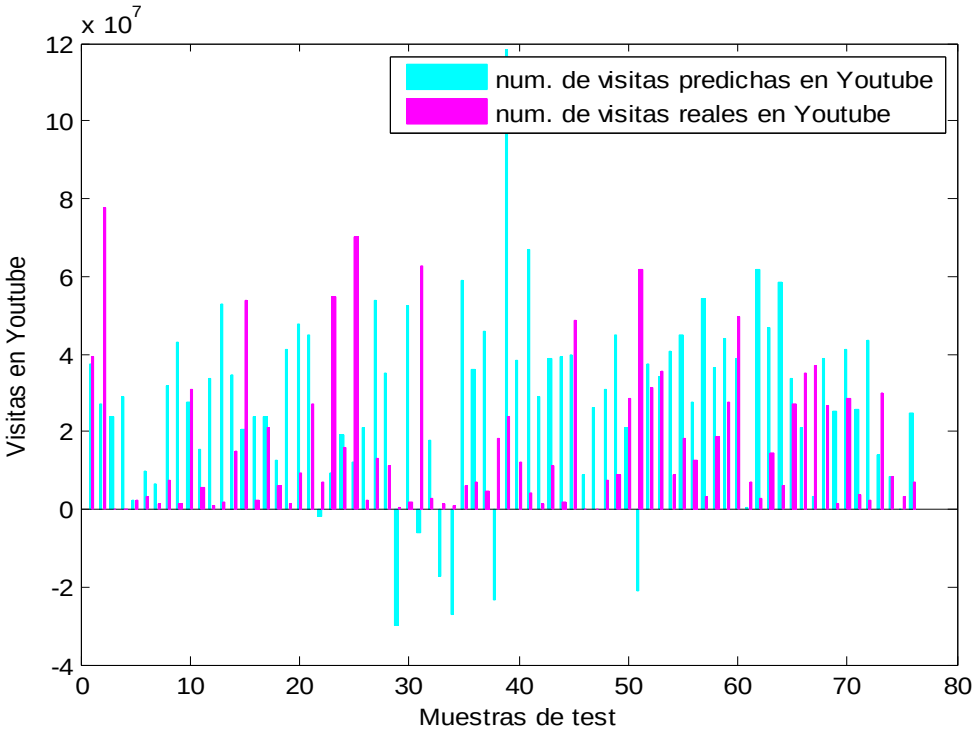


Figura 44. Numero de Visita predichas y originales de muestras de test

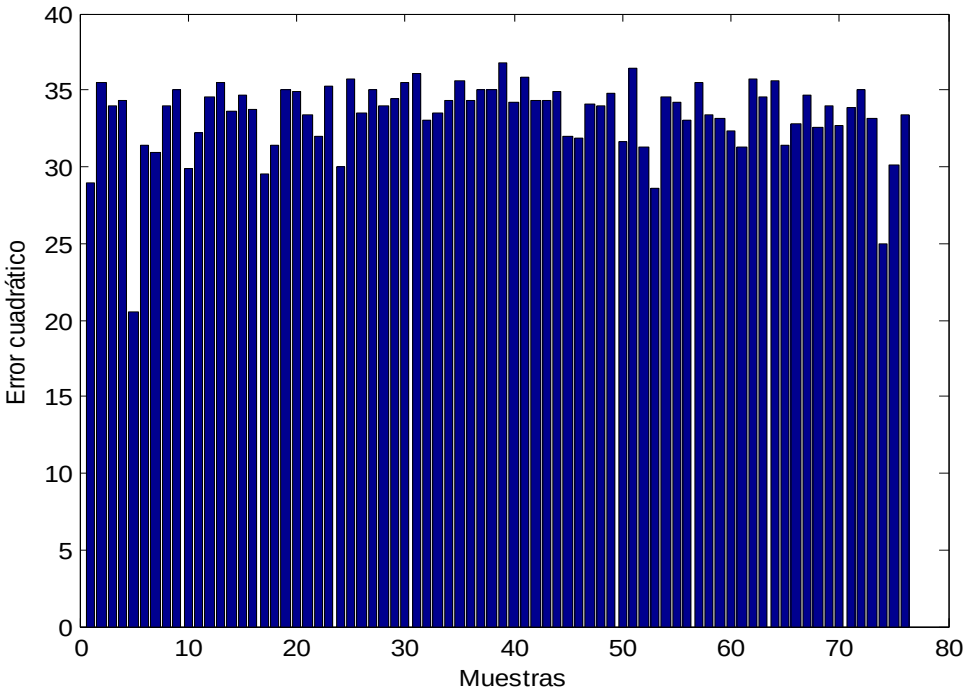


Figura 45. Logaritmo del error cuadrático por muestra

4.2.3.2 Modelo Lineal.Stepwise

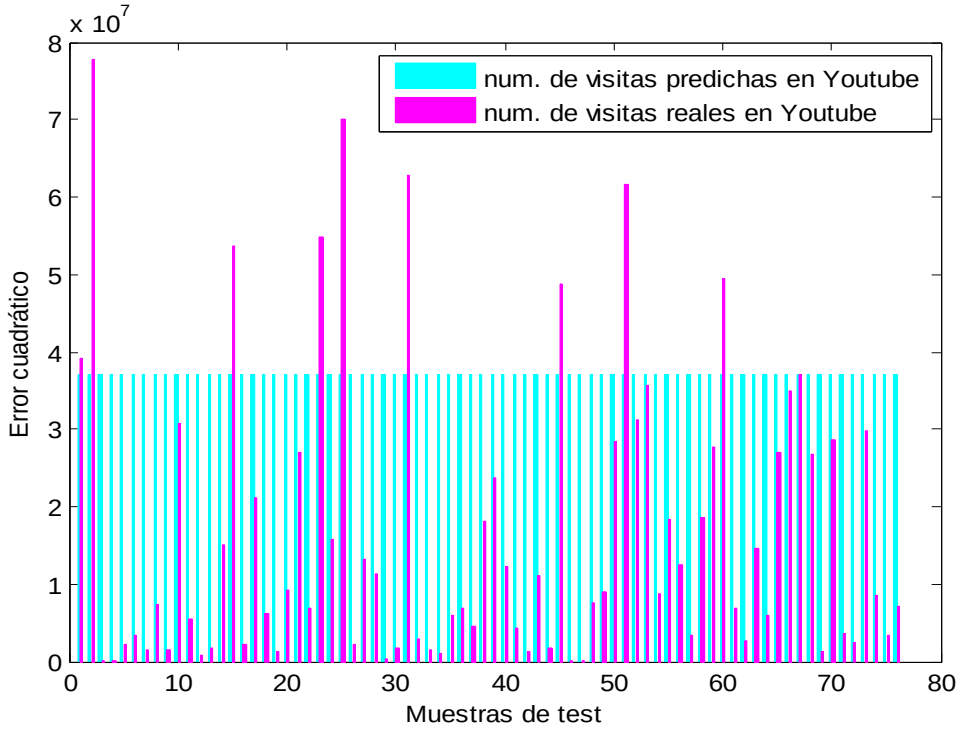


Figura 46. Numero de Visita predichas y originales de muestras de test

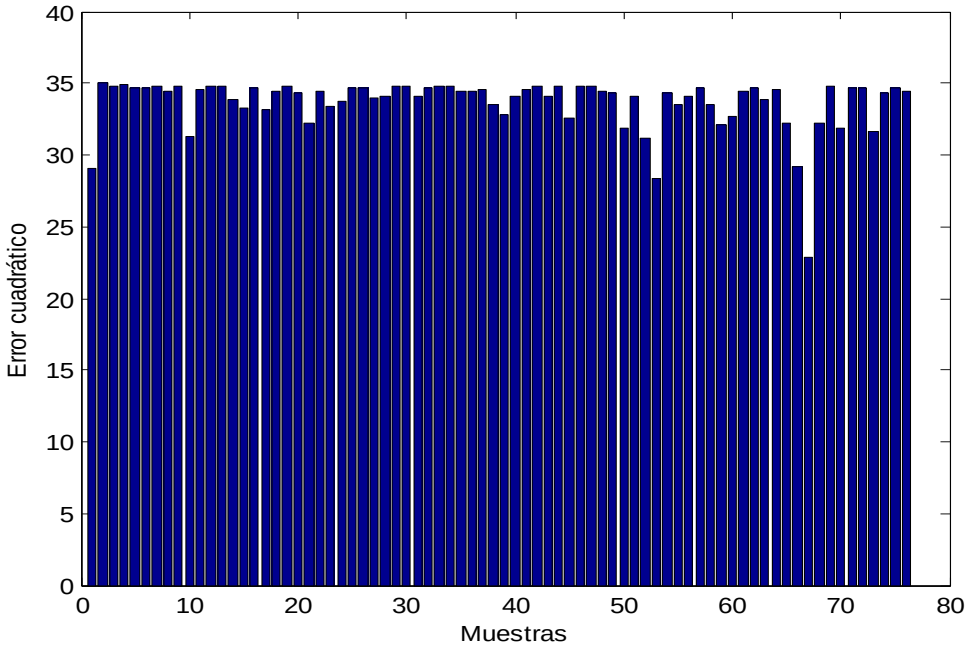


Figura 47. Logaritmo del error cuadrático por muestra

4.2.3.3 Modelo Lineal.Robust

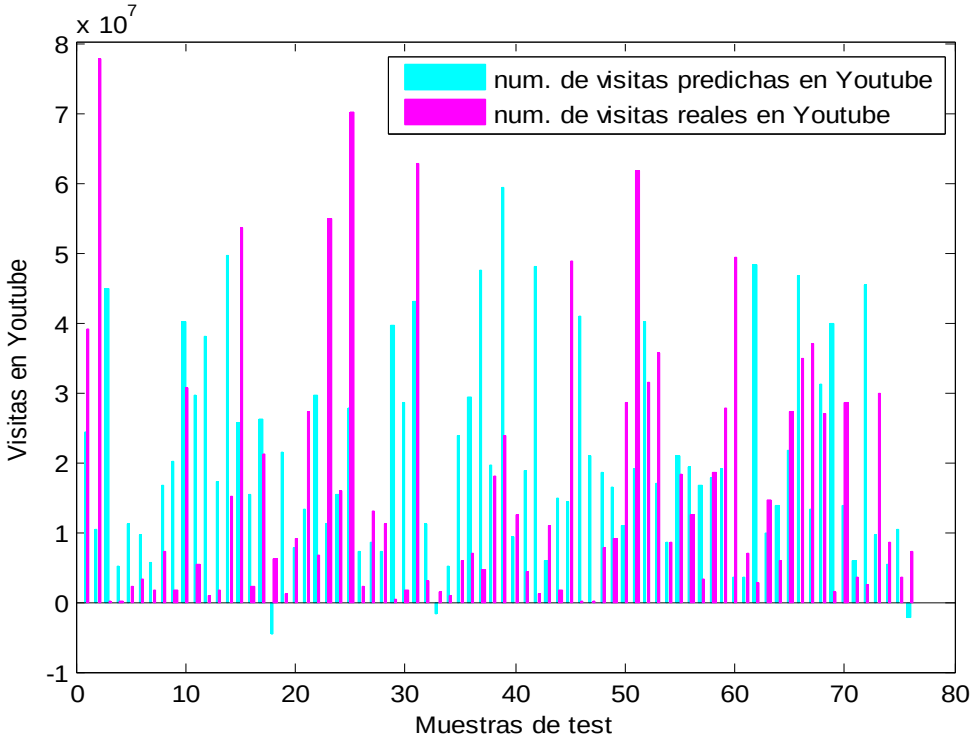


Figura 48. Numero de Visita predichas y originales de muestras de test

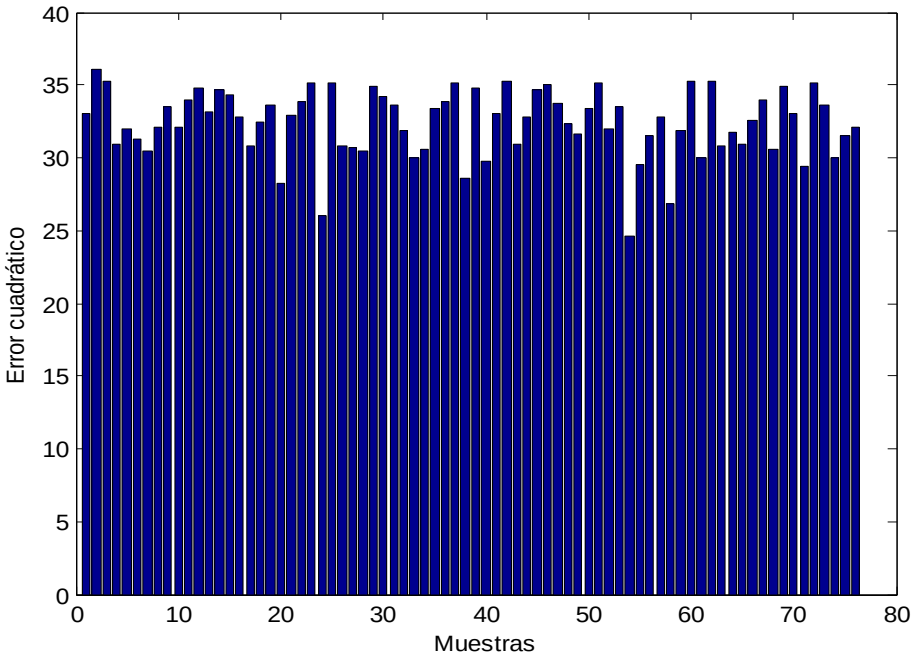


Figura 49. Logaritmo del error cuadrático por muestra

4.2.3.4 Modelo Lasso

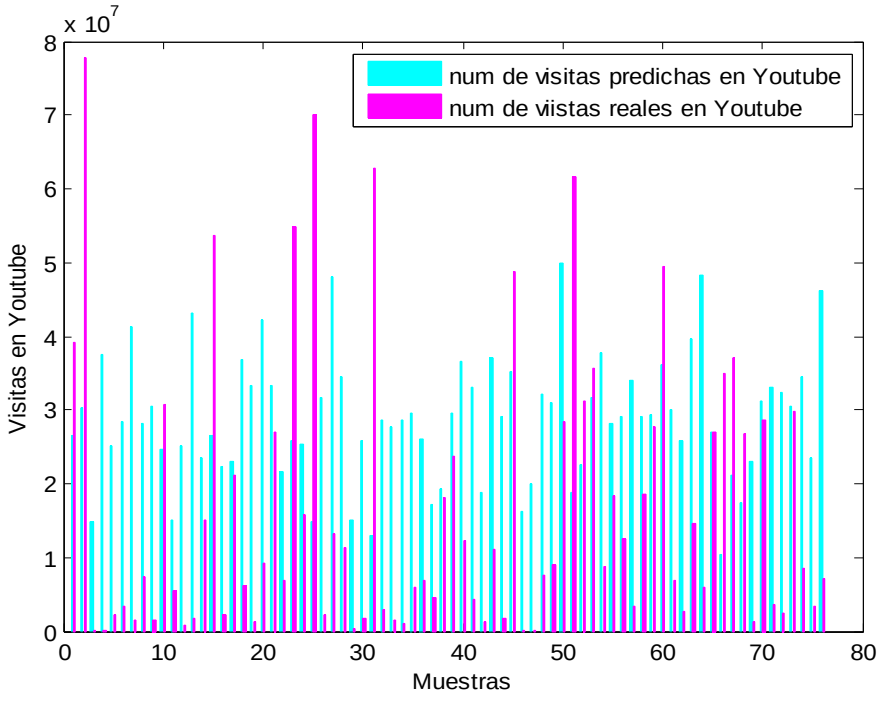


Figura 50. Numero de Visita predichas y originales de muestras de test

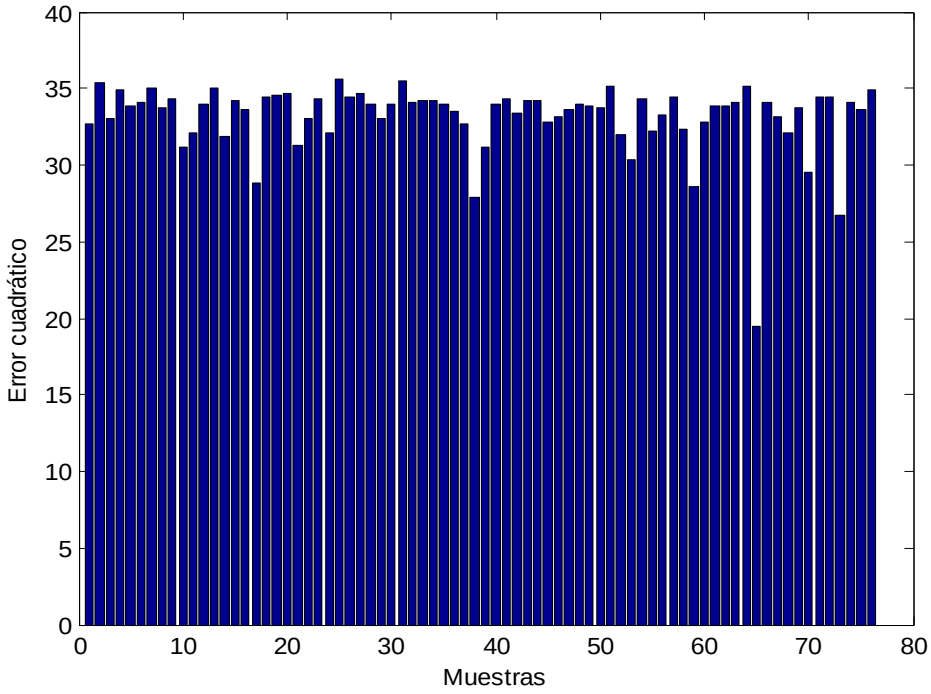


Figura 51. Logaritmo del error cuadrático por muestra

4.2.3.5 Modelo Ridge

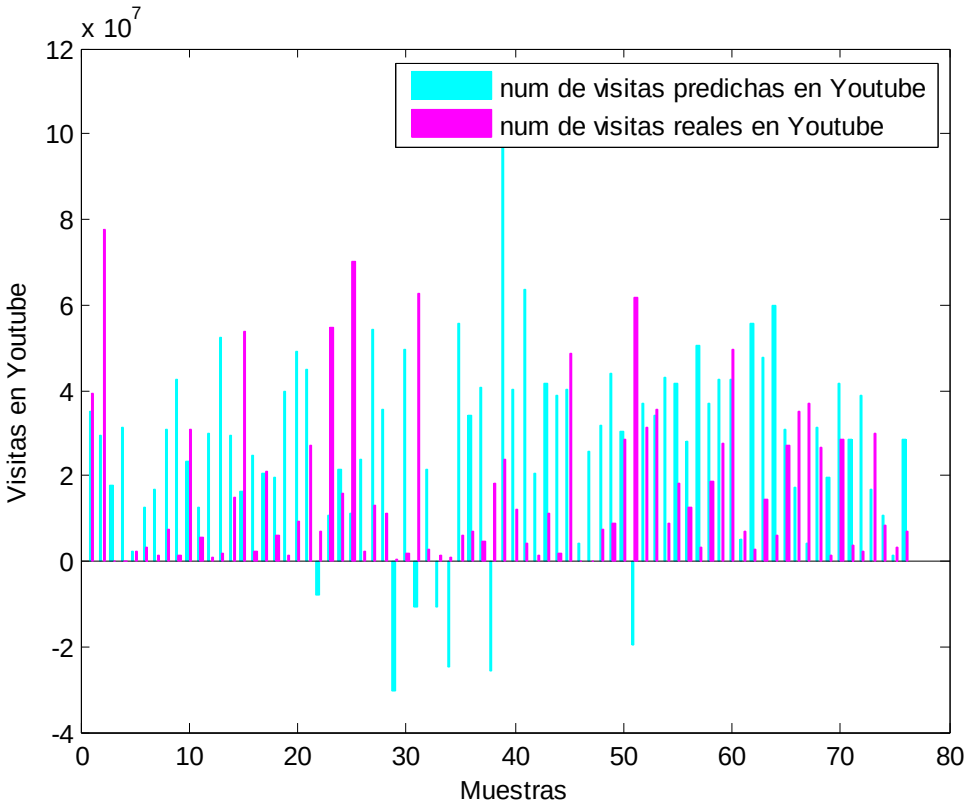


Figura 52. Numero de Visita predichas y originales de muestras de test

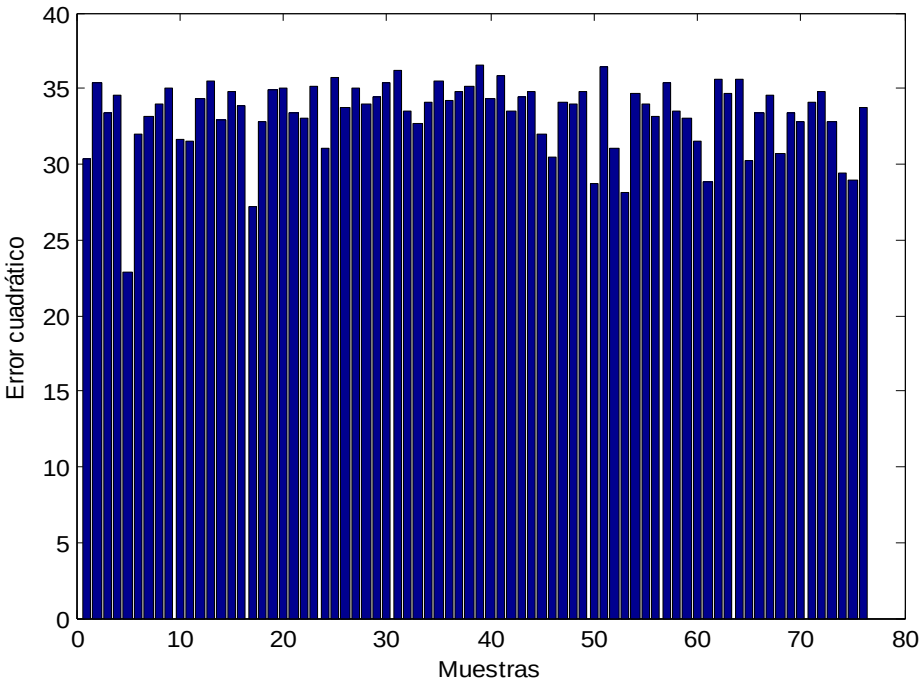


Figura 53. Logaritmo del error cuadrático por muestra

4.2.3.6 Bosques Aleatorios

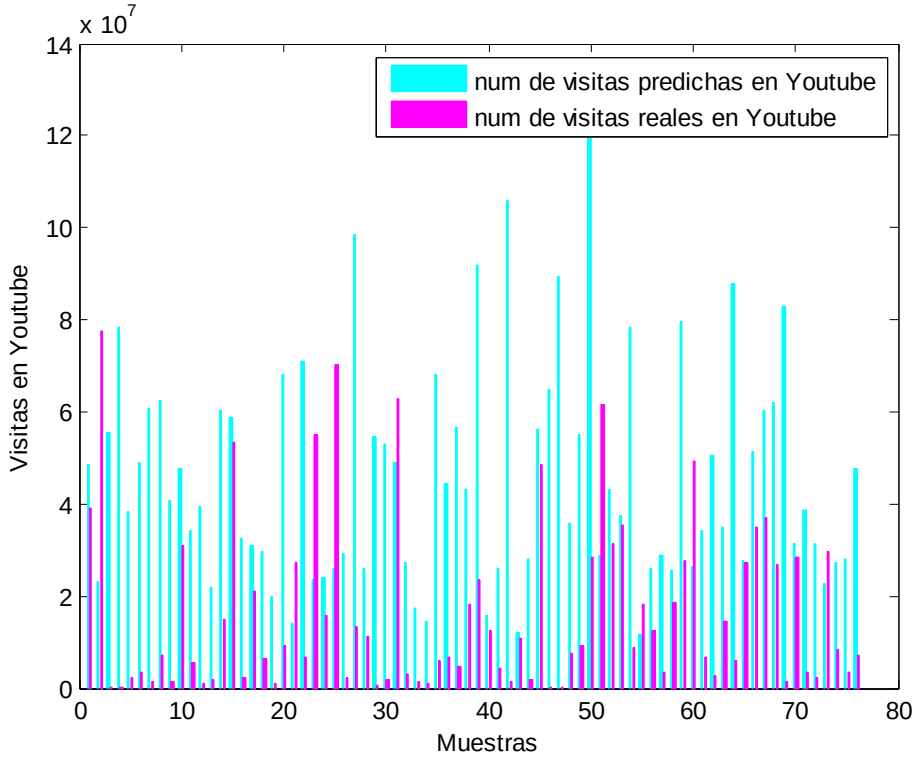


Figura 54. Numero de Visita predichas y originales de muestras de test

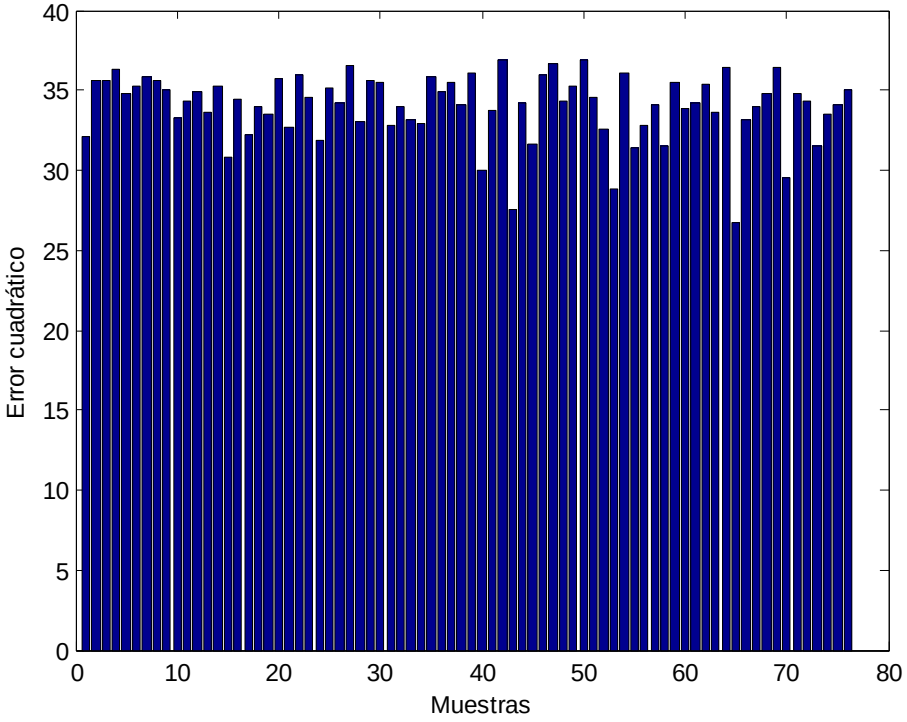


Figura 55. Logaritmo del error cuadrático por muestra

4.3 VALORACIÓN DE RESULTADOS

Las gráficas nos dejan un ligero conocimiento de la forma de predecir del modelo lineal en cada conjunto de entrenamiento de: Rock y Pop. Pero no arroja un valor exacto sobre el error cuadrático medio, para de esta manera asegurarse si sirve o no el modelo para predecir canciones de diferentes géneros musicales.

Es por esto que se procede a la obtención del MSE de cada conjunto de test:

Modelo	*MSE_ POP*	*RMSE_POP*	*MSE_ROCK*	*RMSE_ROCK*
TreeBagger	1.30e+17	3.61e+08	1.86e+15	4.31e+07
Lasso	1.37e+17	3.71e+08	8.67e+14	2.49e+07
Stepwise	1.38e+17	3.72e+08	7.72e+14	2.78e+07
Lineal	1.42e+17	3.77e+08	1.10e+15	3.32e+07
Ridge	1.41e+17	3.75e+08	1.10e+15	3.32e+07
Robust	1.43e+17	3.78e+08	5.50e+14	2.34e+07

Tabla 15. Tabla de MSE de muestras de validación de ambos tipos de música.

En términos generales se puede afirmar que el método de bosques aleatorios (TreeBagger) es el que mejores resultados predice en Pop entre los modelos que se contemplaron. Sin embargo no es el que mejor predice para el Rock.

Nótese que el modelo ridge y regress dan resultados idénticos. Ambos comparten la misma calidad del modelo y además predicen ciertas visitas de canciones, usadas en el conjunto de entrenamiento, con signo negativo. La causa se otorga a la existencia de outliers. Para evitarlo habría que volver a entrenar el modelo quitando estas muestras que no se ajustan a él, sin embargo esto no se ha tomado en cuenta dado que al eliminar muestras para ajustarlas al modelo se disminuye la probabilidad de predecir correctamente nuevos datos atípicos a dicho modelo predictivo.

Si se observan con detenimiento los resultados de la tabla, la diferencia entre valores predichos y reales de Rock son mucho menores que para el Pop. Hay que tener en cuenta que el grupo de test que se utilizó para validar los modelos en la clase Pop, tiene 9 muestras solamente, cuando para Rock hay 76, esto influye en el MSE medio considerablemente y por consiguiente a la desviación del error.

En lo que no cabe duda, es al afirmar que el modelo de bosques aleatorios es el que minimiza el porcentaje de error de predicción a la hora de emplearlo en canciones de género pop. Y para el caso de canciones rock, el uso de modelos lineal y ridge es lo óptimo.

Tablas de resultados muestras Pop

Canción	Artista	Num visitas reales	Num visitas predichas	Ranking real	Ranking predicho	Error relativo
Gangnam Style ()	Psy	1.147.647.977	77396393,25	1	1	0%
Call Me Maybe	Carly Rae Jepsen	378.984.995	259525965,5	2	2	0%
Somebody That I Used To Know	Gotye	364.146.836	214024301,8	3	3	0%
What Makes You Beautiful	One Direction	311.872.040	172970766,4	4	4	0%
Whistle	fun.	161.410.083	133877889,3	6	5	17%
We Are Young (feat. Janelle Monáe)	Flo Rida	181.077.191	123909710,4	5	6	20%
Live While We're Young	Rihanna	127.395.181	101576934,5	8	7	13%
Diamonds	A$AP Rocky	3.020.316	98576830,86	71	8	89%
Wide Awake	One Direction	136.194.530	86905294,01	7	9	29%
We Are Never Ever Getting Back Together	Justin Bieber	100.817.497	84384643,1	12	10	17%
One More Night	Katy Perry	117.829.473	83857502,4	9	11	22%
As Long As You Love Me	Juicy J	14.540.428	82622850,26	48	12	75%
Good Time	Taylor Swift	108.650.503	71506536,89	10	13	30%
Don't Wake Me Up	Hunter Hayes	1.233.028	71269924,85	81	14	83%
A Thousand Years	Owl City/Carly Rae Jepsen	72.213.266	70893752,4	13	15	15%
Little Things	One Direction	64.314.617	64024744,64	16	16	0%
Give Your Heart A Break	Bruno Mars	63.554.527	60595400,03	18	17	6%
Locked Out of Heaven	Lil Wayne	7.987.563	58040924,7	59	18	69%
The A Team	French Montana	29.332.242	58012992,72	31	19	39%
Some Nights	Christina Perri	68.857.079	57142246,5	15	20	33%
Want U Back	Frank Ocean	7.384.199	54945343,87	61	21	66%
Let Me Love You	Demi Lovato	64.205.069	54717117,14	17	22	29%
Thrift Shop (feat. Wanz)	Miguel	16.370.228	54560431,5	45	23	49%
Hall of Fame	Chris Brown	71.075.655	54111899,54	14	24	71%

Canción	Artista	Num visitas reales	Num visitas predichas	Ranking real	Ranking predicho	Error relativo
Don't You Worry Child	Florida Georgia Line	12.934.699	51421348,27	51	25	51%
Little Talks	AWOLNATION	75.523	47918085,49	98	26	73%
Your Body	Ne-Yo	46.350.970	47245034,3	22	27	23%
Too Close	Macklemore & Ryan Lewis	46.099.064	46952807,52	23	28	22%
Girl On Fire	Pitbull	15.723.540	46292871,05	47	29	38%
Blow Me (One Last Kiss)	The Lumineers	28.843.471	45125766,88	32	30	6%
Canción	Artista	Num visitas reales	Num visitas predichas	Ranking real	Ranking predicho	Error relativo
Pop That	Cher Lloyd	47.407.111	44013357,10	21	31	48%
Ho Hey	The Script	44.106.217	43653836,24	24	32	33%
Mercy	Swedish House Mafia	42.242.416	42194775,79	25	33	32%
I Cry	fun.	50.721.138	42141300,78	20	34	70%
Va Va Voom	Ed Sheeran	50.993.887	38995341,76	19	35	84%
Birthday Song	Alicia Keys	32.983.023	38579255,78	29	36	24%
Sweet Nothing	Christina Aguilera	40.198.172	37850995,44	27	37	37%
I Knew You Were Trouble	Alex Clare	38.363.745	37117348,57	28	38	36%
Wanted	Flo Rida	25.974.725	36914711,05	34	39	15%
I Won't Give Up	P!nk	30.012.405	34527746,4	30	40	33%
Clique	Nicki Minaj	25.557.853	34433423,94	35	41	17%
Begin Again	Kanye West	22.600.101	32281446,30	41	42	2%
Madness	T.I./Lil' Wayne	3.732.170	32155229,11	67	43	36%
Ready Or Not	Bridgit Mendler	18.951.720	32044152,75	44	44	0%
Adorn	Brad Paisley	327.502	31557311,31	92	45	51%
Finally Found You	Kanye West	27.767.702	30586859,65	33	46	39%
Don't Stop the Party (feat TJR)	T.I.	665.130	30095071,22	85	47	45%
Bandz A Make Her Dance	2 Chainz	25.064.323	29725295,16	36	48	33%
I Will Wait	Ke$ha	98.149	29193793,57	96	49	49%
Anything Could Happen	Christina Perri	2.473.361	29187346,52	76	50	34%
Cruise	OneRepublic	6.645.445	27349707,26	63	51	19%
Oath	Of Monsters and Men	41.738.413	26308674,78	26	52	100%
Everybody Talks	Taylor Swift	21.988.668	25425758,72	42	53	26%
Rest Of My Life	Ellie Goulding	13.655.149	25127233,1	50	54	8%
Home	Taylor Swift	24.525.589	22904086,53	38	55	45%
Pontoon	David Guetta	7.106.731	22751374,31	62	56	10%
Wicked Games	Muse	19.850.881	21997145,66	43	57	33%

Canción	Artista	Num visitas reales	Num visitas predichas	Ranking real	Ranking predicho	Error relativo
Blown Away	Jason Mraz	23.095.839	21960356,04	40	58	45%
No Worries	The Weeknd	9.849.200	21257149,74	57	59	4%
Skyfall	One Direction	3.194.306	21241102,48	70	60	14%
Thinkin Bout You	Little Big Town	320.071	20919411,05	93	61	34%
Titanium (feat. Sia)	Hunter Hayes	23.627.083	20795079,3	39	62	59%
Feel Again	Imagine Dragons	208.849	20469240,91	94	63	33%
Try	P!nk	6.476.648	20363225,46	64	64	0%
Catch My Breath	Calvin Harris	24.628.112	20242279,14	37	65	76%
Swimming Pools [Drank]	Phillip Phillips	10.676.107	20167846,88	55	66	20%
Ball	Enrique Iglesias/Sammy Adams	15.794.402	19410542,17	46	67	46%
Canción	Artista	Num visitas reales	Num visitas predichas	Ranking real	Ranking predicho	Error relativo
Red	The Band Perry	459.293	19245622,64	87	68	22%
Payphone	Kelly Clarkson	5.768.921	18984979,82	65	69	6%
Kiss You	Imagine Dragons	333.898	18929347,15	91	70	23%
F*ckin Problem	Kendrick Lamar	3.852.693	18022632,39	66	71	8%
Come Wake Me Up	Maroon 5	3.195.004	15461700,83	69	72	4%
Hard To Love	Taylor Swift	3.325.182	15387986,96	68	73	7%
Kiss Tomorrow Goodbye	Gary Allan	1.187.873	15386855,74	82	74	10%
50 Ways To Say Goodbye	Carrie Underwood	8.851.844	15001217,32	58	75	29%
A Thousand Years, Pt. 2 (feat. Steve Kazee)	Mumford and Sons	14.363.028	14984858,82	49	76	55%
Beer Money	Cher Lloyd	12.094.677	14854900,13	52	77	48%
Creepin'	J. Cole	467.288	14843226,3	86	78	9%
Til My Last Day	Ludacris	11.159.865	14762708,08	54	79	46%
Take A Little Ride	Lee Brice	2.890.529	14730104,74	73	80	10%
Somebody's Heartbreak	Eric Church	2.326.705	14592457,87	78	81	4%
Every Storm (Runs Out Of Rain)	Luke Bryan	2.563.983	14418363,72	74	82	11%
Goodbye In Her Eyes	Justin Moore	2.311.746	13262371,58	79	83	5%
The Only Way I Know (with Luke Bryan and Eric Church)	Neon Trees	12.021.993	13126543,36	53	84	58%
Trap Back Jumpin'	Kip Moore	2.439.453	12511352,51	77	85	10%
Miss America	Adele	7.678.528	11239155,34	60	86	43%
Better Dig Two	Randy Houser	409.622	10984062,4	89	87	2%
Just a Fool (feat Blake Shelton)	Justin Bieber	383.681	10911666,09	90	88	2%

Canción	Artista	Num visitas reales	Num visitas predichas	Ranking real	Ranking predicho	Error relativo
How Country Feels	Maroon 5	104.628.243	10888674,42	11	89	100%
Beauty And A Beat	Little Big Town	9.962.785	10682383,39	56	90	61%
Radioactive	Zac Brown Band	852.659	10457860,4	83	91	10%
Southern Comfort Zone	Jason Aldean	1.939.271	8307905,16	80	92	15%
Tornado	Taylor Swift	26.320	8026454,542	99	93	6%
It's Time	Rascal Flatts	2.955.993	6415848,493	72	94	31%
Lights	Christina Aguilera	455.353	6087166,073	88	95	8%
Die Young	Train	2.538.765	5346524,983	75	96	28%
She's Not Afraid	One Direction	84.690	3837247,962	97	97	0%
Sail	Ellie Goulding	161.561	3578662,8	95	98	3%
22	Jason Aldean	842.985	2723140,737	84	99	18%
						Porcentaje de error absoluto medio=36%

Tabla 15. Resumen de predicción de Rock

Song_name	Artist_name	Vistas predichas	Ranking nuevo	Num Visitas	Ranking viejo	Error relativo (%)
'Kryptonite'	'3 Doors Down'	10267856,3	56	77740388	1	5500%
'My Immortal'	'Evanescence'	27666940,1	21	70016206	2	950%
'Sweet Child O'' Mine'	'Guns N'' Roses'	42922411,4	9	62715190	3	200%
'Bohemian Rhapsody'	'Queen'	19037194,1	34	61661535	4	750%
'Bring Me To Life'	'Evanescence'	11078507,9	53	54929026	5	960%
'LIVIN'' ON A PRAYER (Live)'	'Bon Jovi'	25715761,8	23	53592202	6	283%
'Monster'	'Skillet'	3479315,34	73	49385317	7	943%
'Rockstar'	'Nickelback'	14424729,7	46	48692393	8	475%
'Here Without You'	'3 Doors Down'	24366082,9	24	39125564	9	167%
'You Found Me'	'The Fray'	13392226,9	49	36984509	10	390%
'Creep'	'Radiohead'	17007093,3	39	35597371	11	255%
'How To Save A Life'	'The Fray'	46586961,4	6	34912511	12	50%
'We Will Rock You'	'Queen'	40102357,8	11	31339225	13	15%
'Crazy'	'Aerosmith'	40079096,6	12	30751399	14	14%

Song_name	Artist_name	Vistas predichas	Ranking nuevo	Num Visitas	Ranking viejo	Error relativo (%)
'Drops Of Jupiter'	'Train'	9678806,21	59	29827783	15	293%
'Animal I Have Become'	'Three Days Grace'	13911161	47	28650917	16	194%
'Another One Bites The Dust'	'Queen'	10829437,7	54	28508992	17	218%
'Hero'	'Skillet'	19195070	33	27650104	18	83%
'Let It Be'	'The Beatles'	21830960,3	26	27134031	19	37%
'Bad Day'	'Daniel Powter'	13341395,1	50	27123642	20	150%
'IRIS'	'Goo Goo Dolls'	31256997,1	16	26870553	21	24%
'You And Me'	'Lifehouse'	59282349,6	1	23713739	22	95%
'Heaven'	'Bryan Adams'	26287570,5	22	21265575	23	4%
'A Drop In The Ocean'	'Ron Pope'	17878829,6	37	18563096	24	54%
'Under The Bridge'	'Red Hot Chili Peppers'	20888785,6	28	18324619	25	12%
'Stairway To Heaven'	'Led Zeppelin'	19728781,5	31	18082642	26	19%
'Call Me When You''re Sober'	'Evanescence'	15421464,3	44	15861128	27	63%
'Forever Young'	'Alphaville'	49590730,9	2	15139006	28	93%
'Santeria'	'Sublime'	9798423,26	57	14645658	29	97%
'Dreams'	'Fleetwood Mac'	8516589,89	61	13150596	30	103%
'Right Here Waiting'	'Richard Marx'	19301865,1	32	12463006	31	3%
'Sweet Home Alabama'	'Lynyrd Skynyrd'	9456213,88	60	12377384	32	88%
'Go Your Own Way'	'Fleetwood Mac'	7189597,22	65	11306791	33	97%
'Float On'	'Modest Mouse'	5801509,09	67	11045262	34	97%
'Butterfly'	'Crazy Town'	7812577,41	63	9141810	35	80%
'Every Rose Has Its Thorn'	'Poison'	16569851,3	42	9058315	36	17%
'Black Betty'	'Ram Jam'	8460585,65	62	8679930	37	68%
'Headstrong'	'Trapt'	5269630,15	69	8546077	38	82%
'Youth Of The Nation'	'P.O.D.'	18471856,9	36	7663087	39	8%
'Thunderstruck'	'AC/DC'	16816998,5	40	7262792	40	0%
'Play That Funky Music'	'Wild Cherry'	-2233510,87	75	7204763	41	83%
'Don''t Stop Believin'"	'Journey'	29399338,5	19	6902445	42	55%
'Two Princes'	'Spin Doctors'	3502597,95	72	6855328	43	67%

Song_name	Artist_name	Vistas predichas	Ranking nuevo	Num Visitas	Ranking viejo	Error relativo (%)
'MONEY FOR NOTHING (Live)'	'Dire Straits'	29556258,4	17	6829334	44	61%
'Crazy Bitch'	'Buckcherry'	-4710609,68	76	6258630	45	69%
'Imagine'	'John Lennon'	23850288,6	25	6051305	46	46%
'Fly'	'Sugar Ray'	13774464,6	48	5923620	47	2%
'Dream On'	'Aerosmith'	29508778,2	18	5464150	48	63%
'Open Arms'	'Journey'	47439802,9	5	4587279	49	90%
'Come On Get Higher'	'Matt Nathanson'	18891500,5	35	4367970	50	30%
'Chalk Outline'	'Three Days Grace'	6044571,41	66	3645313	51	29%
'Heaven Nor Hell'	'Volbeat'	10398570	55	3433841	52	6%
'HIGHWAY TO HELL (Live)'	'AC/DC'	9678941,36	58	3417526	53	9%
'Lonely No More (Acoustic)'	'Rob Thomas'	16733404,9	41	3286376	54	24%
'I Miss The Misery'	'Halestorm'	11105603,7	51	2977087	55	7%
'Rock''n Me'	'Steve Miller Band'	48223803	3	2790778	56	95%
'You Don''t Know How It Feels (Live)'	'Tom Petty'	45296407,5	7	2396486	57	88%
'Bad Company'	'Five Finger Death Punch'	7247674,98	64	2306110	58	10%
'More Than A Feeling'	'Boston'	15475132,3	43	2256057	59	27%
'Hells Bells'	'AC/DC'	11090955,6	52	2199069	60	13%
'Black Velvet'	'Alannah Myles'	17286766	38	1798774	61	38%
'Photograph'	'Nickelback'	14797756,6	45	1792200	62	27%
'Stars (feat. Kenny Chesney)'	'Grace Potter'	28636773,3	20	1718102	63	68%
'You Shook Me All Night Long'	'AC/DC'	20036749,8	30	1591813	64	53%
'T.N.T.'	'AC/DC'	5775319,45	68	1579772	65	5%
'We Are'	'Hollywood Undead'	-1840208,11	74	1520767	66	12%
'Boys Are Back In Town'	'Thin Lizzy'	39918714,5	13	1363175	67	81%
'Enter Sandman'	'Metallica'	48010102,5	4	1289056	68	94%
'You''re The Inspiration'	'Chicago'	21342761,6	27	1223888	69	61%
'Life''s Been Good'	'Joe Walsh'	5258492,52	70	974445	70	0%
'I Don''t Want To Miss A Thing (Live)'	'Aerosmith'	37895837,7	15	880369	71	79%

Song_name	Artist_name	Vistas predichas	Ranking nuevo	Num Visitas	Ranking viejo	Error relativo (%)
'Feels Like The First Time'	'Foreigner'	39534721,2	14	433976	72	81%
'Crazy Train'	'Ozzy Osbourne'	40822766,8	10	143918	73	86%
'Alive'	'P.O.D.'	20799789,9	29	143918	74	61%
'Back in Black (Live)'	'AC/DC'	44752541,8	8	82680	75	89%
'Dirty Deeds Done Dirt Cheap'	'AC/DC'	5164302,79	71	21518	76	7%
						Error promedio=194%

Tabla 16. Resumen de predicción de Rock

5

Conclusiones

Tras la elaboración de este proyecto se resaltan algunas conclusiones interesantes.

-El prototipo está enfocado en valorar algunas propiedades de las canciones de muestra referentes a la música. Es sumamente importante aclarar este punto dado que los resultados que arroja el predictor no tienen por qué ser exactos al número de visitas reales en Youtube. Fueron descartados aspectos que influyen directamente en el visionado de éstas como lo es la publicidad, la calidad del vídeo, el contenido de éste y el éxito del artista entre otros.

-El género que se escogió para entrenar es un conjunto de muestras complejo. Se explicó con anterioridad que el Pop contempla una amplia gama musical, desde el hip-hop hasta la electrónica. Por lo que conceptualmente sirve para prácticamente todo tipo de música convencional. Para comprobarlo se tomaron 76 muestras de Rock y empleando el mismo modelo. No solo se obtuvo resultados satisfactorios, además predice mejor que con Pop, el error cuadrático medio que arroja es notablemente menor.

- Uno de los objetivos principales del proyecto era conocer las propiedades de la canción que mayor peso se le da a la hora de puntuar un hit. Había dos métodos claros para realizar la selección de características: mediante stepwise o usando lasso. La utilización de stepwise fue un fiasco, dado que el modelo decía que la salida era directamente proporcional al coeficiente independiente, es decir, no depende de ninguna de las 9 variables que se tomaron para el ajuste. En cambio lasso iguala a cero aquellas características que aumentan el error y deja las más relevantes. En el caso de Pop determina que: Tempo, Speechiness, Key, Duration y Danceability son las que definenel éxito. En la tabla 11 de los coeficientes estimados de regresión se puede comprobar lo dicho.

- La propuesta de ranking que se ha desarrollado en este trabajo podría ayudar a valorar cuantitativamente la calidad de cualquier canción comercial. Por lo que con algunas mejoras se podría utilizar como mecanismo de análisis de mercado en la industria musical.

5.1 MEJORAS Y TRABAJOS FUTUROS

Con el fin de potenciar este prototipo a continuación se exponen las siguientes mejoras.

1. Aumentar la base de datos para que se pueda generalizar el modelo y no sea tan específico.
2. Aunque hoy en día se considera que el número de visitas en Youtube sea un indicador de éxito, se puede cambiar la salida del predictor por una valoración cuantitativa del público. De esta manera, los factores independientes a lo musical no influirán de ninguna forma en los pesos, si es que se quiere realizar un estudio de la influencia de las distintas propiedades armónicas en la concepción actual de éxito.
3. Hacer un modelo para cada género musical, se sabe que no todas persiguen las mismas cualidades musicales. De esta manera se podría saber en qué característica centrarse a la hora de componer cuando se busca un éxito en cualquier tipo de música.
4. Añadir características diferentes a lo relacionado con la música. Dado que se ha comprobado que el éxito y por lo tanto el consumo de las canciones, no sólo depende de la calidad sonora y armónica, sino de propiedades ajenas a la música. Existen empresas de marketing que venden este tipo de información, y si se pudieran incluir en el trabajo, potenciarían el prototipo a obtener predicciones mucho más fiables.
5. Eliminar aquellas características redundantes y reajustar el modelo.
6. En cuanto a los métodos de aprendizaje máquina, se podrían utilizar métodos más complejos e innovadores como el de *learning to rank*.
7. Se puede hacer un estudio valorando los pesos que se le da a cada predictor en una generación determinada. Conociendo así los gustos de cada sector demográfico según la edad, para enfocar las ventas del producto a cada una de ellas y saber en cual generación una canción tendría mayor éxito.
8. Desarrollo de un software para facilitar al usuario la interacción con el prototipo, pudiendo introducir una canción y obtener el puesto en el que pudiera estar en las listas de ranking del momento.

Presupuesto

COSTES POR MATERIAL

Ordenador HP Pavilion	700,00 €
Licencia básica e individual de MATLAB R2012a de MathWorks para plataforma Windows o Linux	2.150,00 €
Statics Toolbox de Matlab	200,00 €
Licencia Microsoft Office 2010	120,00 €
Herramienta curl	- €
TOTAL	**3.170,00 €**

COSTE HUMANO

Programación del prototipo y pruebas	240,00 €
Diseño del prototipo	30,00 €
Memoria escrita	180,00 €
Consulta de bibliografía	20,00 €
Coste ingeniero/hora	20,00 €
TOTAL	**9.400,00 €**

PRESUPUESTO DEFINITIVO

Costes por material	3.170,00 €
Coste humano	9.400,00 €
Coste TOTAL del proyecto	**12.570,00 €**

Referencias Bibliográficas

Libros, artículos e informes de conferencias

- (Smith, 1976)[Alten 2002]*Audio in Media*. Standley R. Alten. Sixth edition. Editorial Wadsworth/Thomson Learning (Alten, 1976) (Morik, 2004)
- Smith, D. y. (1976). *Applied Regression Analysis*. John Wiley and sons.
- Herrera, E. (1984). *Teoría Musical y Armonía Moderna*.BARCELONA : ANTONI BOSCH.
- Morik, I. M. (2004). Automatic Feature Extraction for Classifying Audio Data. *Jornal Machine Learning Volume 58* , 2-3.
- Breebaart, M. F. (2003). Features for Audio and Music Classification. *ISMIR*. Eindhoven. (Gérard Biau, 2008)
- Gérard Biau, L. D. (2008). Consistency for a Simple Model of Random Forests and Other Averaging Classifiers. *The Journal of Machine Learning Research,Volume 9*, 2015-2033.
- Wiener, A. L. (2002). Classification and Regression by random Forest. *R News Vol 2/3*, 18-22.(Rashwan, 2011)
- Rashwan, M. E.-D. (2011). Multicollinearity Problem Using Ridge Regression Models. *Int J.Contemp.Math Sciences Vol.6* , 585-600.

- Int. J. Contemp. Math. Sciences Vol. 6, no. 12, 585 – 600 (2011). *Solving* (Castro, 2012)

- *Multicollinearity Problem Using Ridge Regression Models*. Faculty of Commerce, Egypt:M. El-Dereny and N. I. Rashwan.
- Castro, S. (2012). Análisis de datos en Grandes Dimensiones. Estimación y Selección de variables en regresión. *Jornadas Académicas de la Facultad de Ciencias Económicas y Administración.*Santiago de Chile.
- *Departamento de Estadística de la Universidad del estado dePennsylvania,* (VIBERG, 2009)(Resenberg, 1999)
- Resenberg, A. (1999). *Linear Regression with Regularization -CSC 84020-Machine learning.* New York.

- Viberg, M. (2009). *Regularization in Linear.* Göteborg, Sweden.
- Salford Systems (*2002). Random Forests* de la revista *Machine Learning,Wald I.* Boston,USA: Leo Breiman y AdeleC(Wiener, 2002)utler.
- *Czech Technical University* .(University). República Checa: Jiří Franc.

Páginas web

- http://es.wikipedia.orgObtenido en la Red Mundial en Junio 20 de 2012
- *http://echonest.com/*Obtenido en la Red Mundial enAgosto 15 de 2012
- *http://www.mathworks.es/es/help/stats/treebaggerclass.html*Obtenido en la Red Mundial en Enero 2 de 2013
- *http://prtools.org/*Obtenido en la Red Mundial enNoviembre 26 de 2013
- *http://jsonlint.com/* Obtenido en la Red Mundial enNoviembre 26 de 2012
- *http://curl.haxx.se/*Obtenido en la Red Mundial enSetiembre 30 de 2012
- *http://php.net/manual/es/book.curl.php*Obtenido en la Red Mundial en Setiembre 30 de 2012
- *http://musically.com/2011/11/07/the-echo-nest-introduces-speechiness-analysis-attribute*. Obtenido en la Red Mundial en Julio 7 de 2012

Anexos

El siguiente código en Matlab es la implementación del método Top_100_features() para la extracción de las muestras del servidor de Echonest y su almacenamiento en una matriz. De esta manera se pueden manipular los datos como se quiera.

```
function y= Top_100_features()

// Definición de la dirección url del servidor Echonest, pidiéndole el análisis de audio y la puntuación de song_hottness.

google_search='http://developer.echonest.com/api/v4/catalog/read?api_key=5LDUIEIRC4DCDFU1B&format=json&results=100&id=CAWNCCL138856103FC&bucket=audio_summary&bucket=song_hotttnesss';

// Conversión de los datos devueltos por Echonest en formato JSON a estructura.

mat_lecture = parse_json(urlread(google_search));

// Recorrido de la matriz, ordenando los datos y pasándolos de estructura a cell.

For i=1:98
feature(i)= mat_lecture{1,1}.response.catalog.items{1,i}.audio_summary;
end
for j=1:98
   feature_cell(1,j)= (feature(1,j).energy);
   feature_cell(2,j)= (feature(1,j).tempo);
   feature_cell(3,j)= (feature(1,j).speechiness);
   feature_cell(4,j)= (feature(1,j).key);
   feature_cell(5,j)= (feature(1,j).duration);
   feature_cell(6,j)= (feature(1,j).liveness);
   feature_cell(7,j)= (feature(1,j).mode);
   feature_cell(8,j)= (feature(1,j).loudness);
   feature_cell(9,j)= (feature(1,j).danceability);
end
```